

TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Tietoliikennetekniikka
Onni Manninen

Tutkintotyö

ZigBee-verkon luominen Matrix Multimedian E-Blocks opetus-
sarjalla

Työn ohjaaja Kaj Sundström
Työn tilaaja TAMK konelaboratorio
JJJ-Automaatio Oy
Tampere 3/2010

TIIVISTELMÄ

Tutkintotyön tarkoituksena oli tutustua Matrix Multimedian E-Blocks -opetussarjaan, Flowcode-ohjelmointityökaluun sekä luoda demolaitteisto käyttäen hyväksi E-Blocks sarjan ZigBee-kortteja. Työn tarkoituksena oli selvittää, kuinka E-Blocks -opetussarjan laitteilla muodostetaan toimiva ZigBee-verkko eri laitteiden välille, tutkia kuinka laitteiden välillä lähetetään informaatiota sekä arvioida laitteiston luotettavuutta.

E-Blocks -opetussarja toimii muokattavana laitteistoalustana erilaisille PIC- sekä AVR-mikrokontrollereille. Mikrokontrollerit ja niihin liitetyt laitteet ohjelmoidaan Matrix Multimedian omalla Flowcode-ohjelmointityökalulla helposti omaksuttavalla vuokaavio-ohjelmoinnilla. Flowcode huolehtii ohjelmoidun vuokaavion kääntämisestä ensin C-kielelle, sitten assembly-konekielelle ja lopuksi mikrokontrollerin ymmärtämäksi heksakoodiksi. Lopullisen ohjelman Flowcode siirtää mikrokontrollerille USB-liitännän kautta.

Matrix Multimedian E-Blocks -sarjaan kuuluu lukuisia yhteensopivia lisäkortteja, kuten LCD-näyttöjä, painonappikortteja, näppäimistöjä, sensorikortteja, relekortteja tai esimerkiksi langattomaan tiedonsiirtoon liittyviä lisäkortteja. Tutkintotyössä käytettiin E-Blocks -sarjaan kuuluvia ZigBee-kortteja sekä muita demolaitteiston tarvitsemia kortteja.

Demolaitteiston halutut toiminnot määritteli koneautomaation laboratorioinsinööri Seppo Mäkelä. Tutkintotyön varsinaiseksi tehtäväksi muodostui Flowcode-ohjelman käytön opettelu, demolaitteiston kokoaminen sekä ohjelmoiminen ja laitteistolla muodostetun ZigBee-verkon luotettavuuden analysointi.

Työn tuloksena voidaan todeta, että Matrix Multimedian tarjoaman ohjelmointityökalun sekä E-Blocks -sarjan avulla voi suunnitella sekä toteuttaa vaativia laitteistokokonaisuuksia kohtuullisen lyhyen perehtymisen jälkeen. Laitteiston avulla voi helposti tutustua mikrokontrollerien tarjoamiin käytännön mahdollisuuksiin, sekä valmistaa hyötysovelluksia esimerkiksi kotiautomaation käyttöön.

Information technology

Telecommunications engineering

Writer Onni Manninen

Thesis Creating ZigBee network with Matrix Multimedia E-Blocks
ZigBee solution pack

Pages 42

Graduation time 3/2010

Thesis supervisor Kaj Sundström

Co-operating company TAMK University of Applied Sciences, JJJ-Automaatio Oy

ABSTRACT

This engineering thesis is about Matrix Multimedia Flowcode program and how to create ZigBee network with Flowcode and E-Blocks ZigBee solution pack. The main assignment was to research how to send information between coordinator and end-devices and evaluate reality of the ZigBee solution pack.

E-Blocks is compatible with PIC and AVR microprocessors and they can be programmed easily with Flowcode. Flowcode translates programs to C++, then from C++ to assembly and last to hexcode. Flowcode also transfers hexcode to microprocessor via USB.

Matrix Multimedia also offers large range of E-Blocks compatible hardware like LCD-screen, switch board, relay board and led board. There are also wireless hardware like RFID board, bluetooth board and ZigBee board. In this thesis ZigBee boards and other E-Blocks compatible hardware was used when demo system was designed, created and programmed.

The result of this work was great success. Demo system worked as designed and my personal knowledge about microprocessors, ZigBee networks, Flowcode and E-Blocks hardware was dramatically increased.

ALKUSANAT

Tutkintotyö kirjoitettiin keväällä 2010 Tampereen ammattikorkeakoulun konelaboratorion sekä JJJ-Automaatio Oy:n pyynnöstä. Tutkintotyön alustavat tutustumiset käytettyyn ohjelmointityökaluun aloitettiin keväällä 2009 ja lopullinen työssä käytetty laitteisto saatiin tammikuussa 2010. Työn vaatima panos tutkimisen, uuden ohjelmiston toiminnan opetteluun sekä demolaitteiston ohjelmointi lisäsi runsaasti tietämystäni ZigBee-verkoista sekä mikrokontrollerien käyttömahdollisuuksista.

Haluan kiittää Tampereen ammattikorkeakoulun konelaboratoriota käytetystä laitteistosta, laboratoriotilan vapaasta käytöstä sekä laboratorioinsinööriä Seppo Mäkelää ohjeistuksesta ja opastuksesta.

Tampereella maaliskuussa 2010

Tämä tutkintotyö on julkinen.

Onni Manninen

SISÄLLYS

	Sivu
1 JOHDANTO.....	1
2 E-BLOCKS -OPETUSSARJA.....	2
2.1 Työssä käytetyt kortit	3
2.2 PIC-16F788A prosessori	7
3 FLOWCODE	10
3.1 Flowcoden työkalut	10
3.2 Esimerkkiharjoitukset	11
3.2.1 Ennakkovalmistelut	11
3.2.2 LCD-näytölle kirjoittaminen	12
3.2.3 Ledien sytytys painonapeista	13
3.3 Zigbee-verkon luominen ja hyödyntäminen	14
3.3.1 Koordinaattori	14
3.3.2 Päätelaitte	16
4 ZIGBEE-JÄRJESTELMÄN SUUNNITTELU, TOTEUTUS JA ANALYSOINTI.....	18
4.1 Miksi ZigBee?	18
4.2 Järjestelmän suunnittelu.....	20
4.3 Käytettävän laitteiston mahdollisuudet ja rajoitukset.....	20
4.4 Järjestelmän toteutus.....	21
4.4.1 Koordinaattorin toteutus.....	22
4.4.2 Oven turvakoodin toteutus	23
4.4.3 Oven moottorionjauksen toteutus	24
4.5 Binaariluvun lähetys koordinaattorille	26
4.6 Lämpötila-anturin toteutus.....	27
4.7 Verkon kantomatka, luotettavuus sekä analysointi.....	28
4.7.1 Verkon kantaman mittaukset.....	28
4.7.2 Laitteiston luotettavuus ja havaitut ongelmat	28
4.8 Zigbee-pakettien analysointi ZENA-analysaattorilla	30
5 LOPPUSANAT	31
LÄHTEET	32
LIITTEET	34
Liite 1: ZigBee-verkon muodostus E-Blocks -sarjalla	34
Liite 2: Koordinaattorin vuokaavio	38
Liite 3: Langattoman näppäimistön vuokaavio	39
Liite 4: Oven moottorionjauksen vuokaavio	40
Liite 5: Binaariluvun lähettäjän vuokaavio.....	41
Liite 6: Lämpötila-anturin vuokaavio	42

LYHENTEIDEN JA TERMIEN SELITYKSET

A/D-muunnin	Mikrokontrollerissa sijaitseva analogia-digitaalimuunnin, joka muuntaa jatkuvan analogisen signaalin digitaalseksi numeroarvoksi.
ALU	ALU, aritmeettis-looginen yksikkö, on PICmikron sydän, jonka läpi kaikki PICmikrossa kulkeva tieto kulkee.
BPSK	Binary Phase-Shift Keying. 868MHz sekä 915MHz taajuudella toimivan ZigBee-signaalin modulaatiotapa, jossa kahdella kanta-aallon eri vaiheella ilmaistaan binäärisen viestin arvo 0 tai 1.
Coordinator	Katso koordinaattori.
CSMA/CA	Carrier Sense, Multiple Access / Collision Avoidance, Zigbee-standardin käyttämä pakettien törmäyksenestomenetelmä. Lähettäjä kuuntelee radiorajapintaa ja lähettää vasta kun radiorajapinta on vapaa.
E-Blocks	Matrix Multimedian tarjoama opetussarja, johon kuuluu erilaisia mikroprosessoreita tukeva multiprogrammer, sekä laaja kirjo siihen liitettäviä sensorikortteja.
EEPROM	Electronically Erasable Programmable Read-Only Memory. Haihtumatonta puolijohdemuistia, jolle voidaan uudelleenkirjoittaa noin 10 000-100 000 kertaa.
End-device	Katso päätelaite.
Flash	Puolijohdemuistityyppi, joka voidaan tyhjentää ja uudelleenohjelmoida sähköisesti. Flash-muisteissa säilyy tieto vaikka muistista kytketään virrat pois.
Flowcode	Matrix Multimedian kehittämä mikrokontrollerin ohjelmointityökalu.
gLCD	Graphical Liquid Crystal Display, graafinen värinäyttö.
Kortti	Multiprogrammeriin yhdistettäviä lisäkortteja kuten sensoreita, näyttöjä, painonappeja yms. kutsutaan korteiksi.
Koordinaattori	<i>Coordinator</i> , ZigBee-verkon isäntä, joka vastaa verkon käynnistämisestä ja johon muut ZigBee-verkon laitteet ovat yhteydessä.
LCD	Liquid Crystal Display, mustavalkoinen kaksirivinen näyttö.

LED	Light Emitting Diode, valoa lähettävä diodi.
Mikrokontrolleri	Mikrokontrolleri on mikropiiri, jossa on oma mikroprosessori, muistia sekä ulkoiset liitäntäjohdot.
Multiprogrammer	E-Blocks -sarjan keskeisin osa; piirilevy, johon kiinnitetään käytettävä mikrokontrolleri. Käytettävät kortit ovat yhteydessä multiprogrammeriin kytkettyyn mikrokontrolleriin RS323-sarjaporttien kautta.
Node	Verkon päätelaitteita voidaan kutsua myös nodeiksi eli verkon solmuiksi, kts. päätelaite.
OQPSK	Offset Quadrature Phase-Shift Keying. 2,4Ghz taajuudella toimivan ZigBee-signaalin modulointitapa, jossa neljällä eri kanta-aallon vaiheella voidaan ilmaista binäärisen viestin arvoja 00, 01, 10 sekä 11.
PAN	Personal Area Network tarkoittaa samassa osoite-avaruudessa toimivien laitteiden muodostamaa verkkoa.
Päätelaite	<i>End-device</i> , ZigBee-verkon päätelaite, joka on yhteydessä koordinaattoriin. Päätelaitteet toimivat usein paristoilla, ja lähettävät esimerkiksi anturitietoa koordinaattorille.
RAM	Read Access Memory, mikroprosessorin käyttömuisti johon suoritettavat sovellukset ja näiden tarvitsemat tiedot ladataan.
Reititin	<i>Router</i> , ZigBee-verkon reititin, jonka avulla verkon kantama voidaan laajentaa.
Router	Katso reititin.
USART	Universal Synchronous / Asynchronous Receiver-Transmitter, mikrokontrollerissa sijaitseva sarjaliikenteen lähetys- ja vastaanottopiiri. Piiri muuntaa rinnakkaisdatan sarjamuotoiseksi.
USB	Universal Serial Bus, yksi yleisimmin käytetyistä liitäntämuodoista tietokoneen ja oheislaitteen välille.
ZENA	ZENA on ZigBee-verkon analysaattori, joka liitetään tietokoneeseen USB-liitännän kautta. ZENA:n ohjelmiston avulla ZigBee-verkon liikennettä voidaan seurata ja kaapata radiorajapinnassa liikkuvia paketteja tarkastelua varten.
ZigBee	Langaton IEEE 802.15.4 –standardin mukainen lyhyen kantaman tietoliikenneverkko, joka toimii 868MHz, 915MHz tai 2,4GHz taajuudella.

1 JOHDANTO

Matrix Multimedia on britannialainen vuonna 1993 perustettu teknologiayritys, jonka päämaja sijaitsee Halifaxissa, Englannissa. /1/ Yritys tarjoaa sulautettujen järjestelmien työkaluja yritysten, oppilaitosten, opiskelijoiden sekä yksityisten asiakkaiden käyttöön. Matrix Multimedian tuoteperheeseen kuuluu mm. mikroprosessorien ohjelmointiin tarkoitettu graafinen vuokaavio-ohjelmointityökalu Flowcode sekä helposti muunneltavissa oleva lisäkortteihin perustuva E-Blocks -opetussarja. Yritys tarjoaa myös erilaisia Flowcode -yhteensopivia sensorikortteja. /2/

E-Blocks -opetussarja sekä Flowcode-ohjelmointityökalu on ollut osana Tampereen ammattikorkeakoulun koneautomaation perusopinnoita jo vuodesta 2008. Matrix Multimedian E-Blocks -sarjaan on kehitetty kuitenkin paljon erilaisia lisäkortteja, joita ei ole vielä hyödynnetty koulun opetuskäytössä.

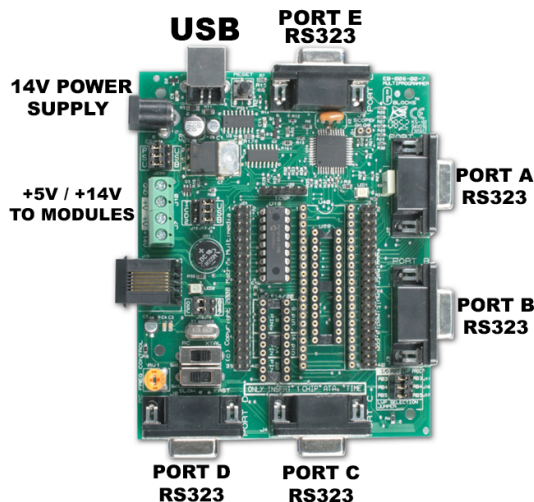
Tutkintotyön tavoitteena oli tutustua E-Blocks -sarjan ZigBee -opetuslaitteeseen, johon kuului koordinaattori sekä kolme päätelaitetta tarvittavine lisäkortteineen. Opetuslaitteistoon kuului myös verkon analysoimiseen tarkoitettu ZENA-analysaattori, sekä tarvittavat ohjelmistot ohjeineen.

Työn tarkoituksena oli selvittää kuinka ZigBee-kortteja ohjelmoidaan Flowcode -ohjelmalla, kuinka korttien välille muodostetaan verkko ja millaisia käytännön hyötykohteita ZigBee -korttien avulla voisi muodostaa. ZigBee -opetuslaitteen lisäksi käytössä oli myös ylimääräisiä ZigBee- sekä muita lisäkortteja.

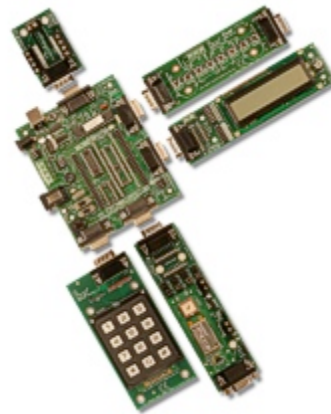
Tutkintotyön lopputulokseksi muodostui viiden erillisen laitteen muodostama verkko, jossa neljä päätelaitetta olivat yhteydessä niin sanottuun keskuslaitteeseen, koordinaattoriin. Laitteistossa oli kolme toimintakokonaisuutta: lämpötilatiedon lähetys koordinaattorille, binääriluvun lähetys tavumuodossa koordinaattorille sekä oven etäavaus mikäli käyttäjä näppäilee oikean nelinumeroisen salasanan yhden päätelaitteen näppäimistöle.

2 E-BLOCKS -OPETUSSARJA

Matrix Multimedian E-Blocks -opetussarja on suunnattu oppilaitoksille ja opiskelijoille erilaisten laitekoonpanojen testausta ja ohjelmointia varten. E-Blocks -sarjan keskeisin laite on multiprogrammeriksi kutsuttu piirilevy, joka tukee useita erilaisia mikrokontrollereita (kuva 1). Multiprogrammer toimii mikrokontrollerien isäntäalustana ja hoitaa virransyötön käytettävälle mikrokontrollerille. Multiprogrammer tarjoaa myös USB-liitännän mikrokontrollerin ohjelmointia varten sekä liitännät tarvittaville lisäkorteille RS323-sarjaporttiliitäntöjen kautta. E-Blocks tukee esimerkiksi PICmicron, AVR:n sekä ARM:n mikrokontrollerituoteperheitä.



Kuva 1. Multiprogrammer /3/



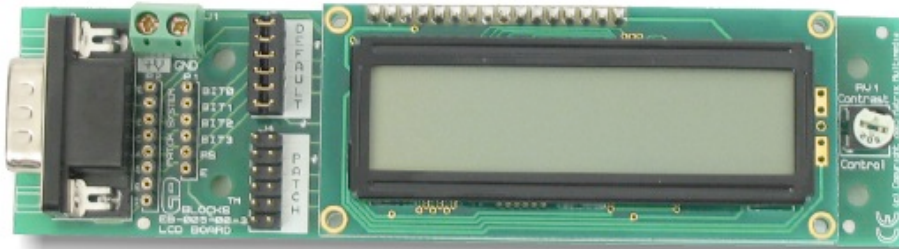
Kuva 2. E-Blocks -sarjan lisäkortteja /4/

E-Blocks -sarja käsittää myös kymmeniä erilaisia multiprogrammeriin sarjaporttiliitännällä yhdistettäviä lisäkortteja (kuva 2), kuten led-valoja, LCD-näyttöjä, relekortteja sekä esimerkiksi tässä tutkintotyössä tutkittuja ZigBee-kortteja. Suurin osa lisäkorteista vaatii ulkoisen virtalähteen toimiakseen ja tarvittavan virran korteille saa multiprogrammerissa sijaitsevista +5V/+14V virtaliittimistä.

E-Blocks -sarjalla on mahdollista luoda erittäin monipuolisia ja eri tarpeisiin soveltuvia laitekokonaisuuksia. Sarjan ydintarkoituksena on mahdollistaa erilaisten demolaitteistojen suunnittelut ja suorittaa tarvittavat testaukset helposti ja nopeasti. Kaikista käytetyistä lisäkorteista on saatavilla myös kattavat piirikaaviot, joiden avulla on laitteistosta mahdollista valmistaa lopullinen sulautettu järjestelmä lopullista käyttötarkoitusta varten.

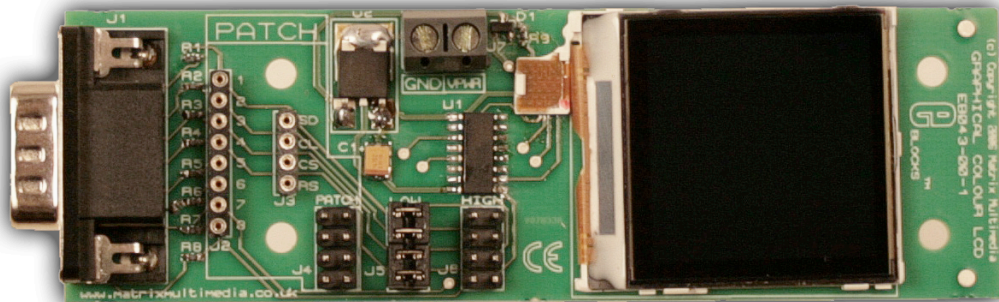
2.1 Työssä käytetyt kortit

Työssä käytettiin viittä eri multiprogrammeria, joihin yhdistettiin useita erilaisia lisäkortteja, riippuen mihin tarkoitukseen mikäkin multiprogrammer oli tarkoitettu. Seuraavassa on esitelty työssä käytetyt kortit.



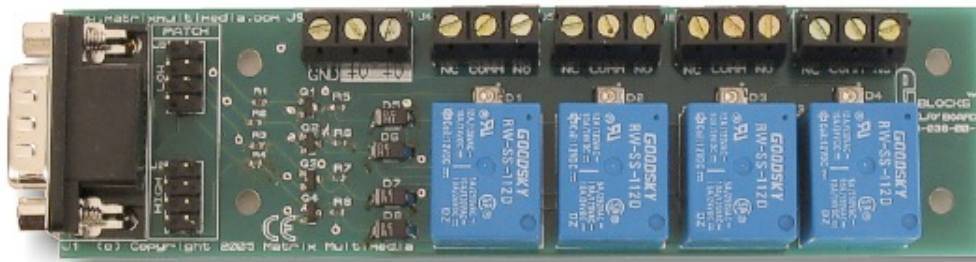
Kuva 3. LCD-näyttökortti /5/

E-Blocks -sarjan LCD-kortti sisältää kaksirivisen mustavalkoisen LCD-näytön ja vaatii toimiakseen 5V jännitteen multiprogrammerilta. Mikrokontrollerin tiedonvälitys LCD-näytölle hoidetaan RS323 –sarjaporttiliitännän kautta. Yhdelle riville mahtuu 16 merkkiä.



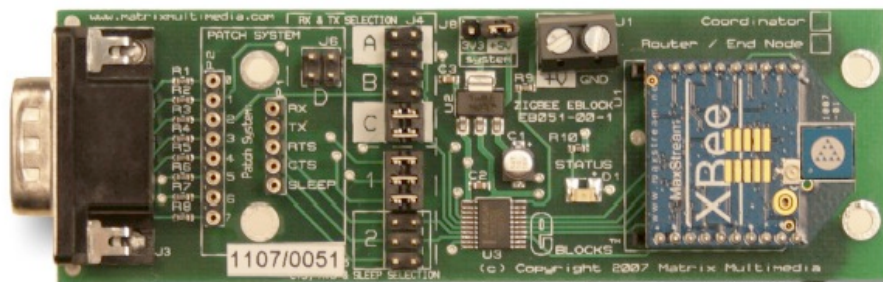
Kuva 4. Graafinen LCD-näyttökortti /6/

E-Blocks -sarjan graafinen LCD-näyttökortti on taustavalaistu 130x130 pikselin värinäyttö, joka kykenee yli 65 000 väriin. Näyttö tukee jopa 16 riviä tekstiä, joten näyttö on huomattavasti monipuolisempi kuin kaksirivinen mustavalkoinen LCD-näyttö. Graafinen LCD-näyttökortti vaatii toimiakseen 14V jännitteen multiprogrammerilta.



Kuva 5. Relekortti /7/

E-Blocks -sarjan relekortti sisältää neljä relettä, joita voi ohjata RS323-sarjaportin kautta. Releillä on kaksi eri tilaa (kiinni/auki) ja relelaatikon ja liitántäruuvien välissä sijaitseva punainen led-valo ilmaisee releen tilan. Relekortti vaatii toimiakseen 14V jännitteen multiprogrammerilta ja releet kestävät maksimissaan 250V jännitteen ja 6A virran.



Kuva 6. ZigBee-kortti /8/

E-Blocks -sarjaan kuuluu kahdenlaisia ZigBee-kortteja: kortteja, jotka toimivat koordinaattoreina ja kortteja, jotka toimivat joko reitittiminä tai päätelaitteina

Koordinaattori on niin sanotusti ZigBee-verkon keskeisin laite, johon reitittimet ja päätelaitteet yhdistävät. Koordinaattorin tehtäviin kuuluu käynnistää ZigBee-verkko sekä päättää, mitkä laitteet saavat yhdistää ja olla yhteydessä muodostettuun verkkoon.

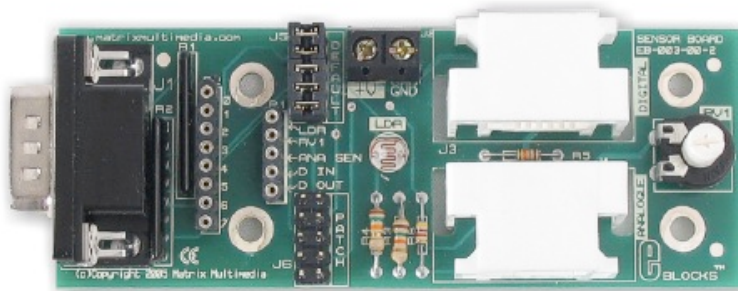
Reitittimiä tarvitaan verkon kantaman laajentamiseen, viestipakettien välittämiseen, ja ne toimivat koordinaattoreiden ja päätelaitteiden välissä. Päätelaitteet toimivat useimmiten informaation kerääjinä lähettäen esimerkiksi anturitietoa koordinaattorille, joko suoraan tai reitittimen välityksellä.

E-Blocks -sarjan ZigBee-kortit käyttävät V2 XBEE-piiriä verkon luomiseen. Piiri tukee vuoden 2007 ZigBee Pro / ZNET standardeja. ZigBee-kortit tarvitsevat toimiakseen +5V jännitteen multiprogrammerilta. ZigBee-kortti on yhteydessä käytettävään mikrokontrolleriin RS323-sarjaporttiliitännän kautta.



Kuva 7. Screw terminal-kortti /9/

E-Blocks -sarjan screw terminal -kortti, eli ruuviliitinkortti, mahdollistaa esimerkiksi kytkimien tilojen lähetyksen mikrokontrollerille. Kortissa on kahdeksan liitäntäruuvia, joihin syötetyt 5V jännitelukemat saadaan muunnettua RS323-sarjaporttiliitännän kautta mikrokontrollerin ymmärtämään 8-bitin tavumuotoon.



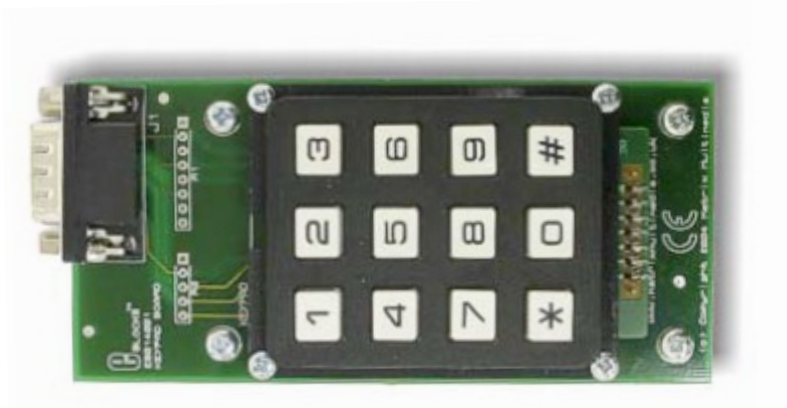
Kuva 8. Sensor board, anturikortti /10/

E-Blocks -sarjan sensor boardin avulla voi opetussarjaan liittää suuren määrän erilaisia antureita. Kortti sisältää kiinteän valosensorin sekä potentiometrin, mutta myös analogisen että digitaalisen sisääntuloliitännän. Kortti tarvitsee toimiakseen ulkoisen +5V jännitteen multiprogrammerilta.

Liitäntöjen kautta voi laitteistoon yhdistää erilaisia antureita, kuten lämpötila-, paine-, kiihtyvyys- sekä liikeentunnistusanpureita. Analogisen sisääntulon jännitevaihtelut muutetaan mikrokontrollerissa A/D muuntimessa analogisesta digitaaliseen muotoon, joten sensor board täytyy yhdistää mikrokontrollerin A/D ominaisuutta tukevaan RS323-sarjaliitäntäporttiin.

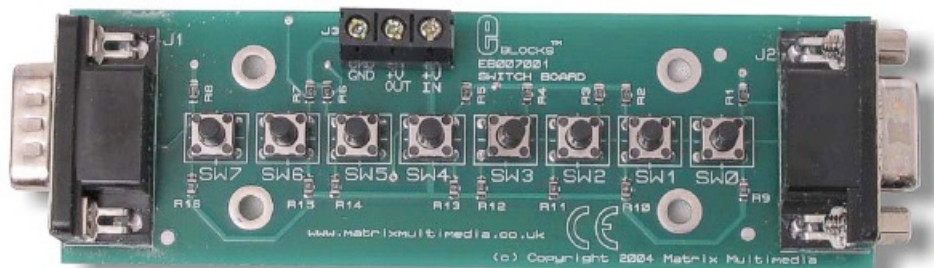
Työssä käytetty lämpötilatieto tuotettiin potentiometrillä mahdollisimman monipuolisen ja vastaavasti tarvittaessa tasaisen lukeman saavuttamiseksi. Teknisen toteutuksen sekä oh-

jelmoinnin kannalta tarkasteltuna ei ollut käytännössä merkitystä tuliko lämpötilatieto simuloituna potentiometriltä vai oikeasta lämpötila-anturista analogisen sisääntulon kautta.



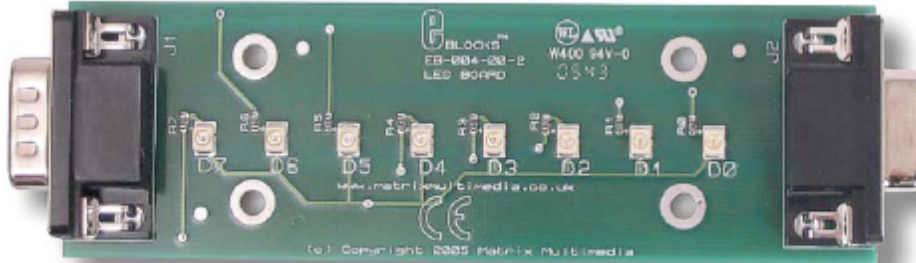
Kuva 9. Numeronäppäimistö-kortti /11/

E-Blocks -sarjan numeronäppäimistö toimii RS323-sarjaporttiliitännän kautta eikä tarvitse erillistä virtaa toimiakseen. Painallukset voidaan tallentaa esimerkiksi ASCII-muotoon ja hyödyntää ohjelman toimintalogiikassa tarpeen mukaan. Näppäimistöä voidaan käyttää hyödyksi esimerkiksi tunnusluvun pyytämisessä tai vaikkapa laitteiston toimintojen ohjaamiseen.



Kuva 10. Switch board, painonappikortti /12/

E-Blocks -sarjan switch board sisältää kahdeksan painonappia, joiden tila kerätään siirrettäen mikrokontrollerille 8-bitin tavumuodossa. Painonapit on nimetty SW0-SW7. Esimerkiksi painonappien SW0, SW1 ja SW4 painuessa pohjaan, tilatieto mikrokontrollerille lähetetään vastaavana binäärilukuna eli tässä tapauksessa 00010011. Kyseinen tavu tarkoittaa kokonaislukuina 1+2+16 eli lukua 19. Painonappikortteja pystyy tarvittaessa myös kettämään ja kortti tarvitsee toimiakseen 5V jännitteen multiprogrammerilta.



Kuva 11. Led-kortti /13/

E-Blocks -sarjan led-kortti on yleisesti harjoituksissa käytetty kahdeksan punaisen ledin kortti, joita ohjataan RS323-sarjaportin kautta. Kortti ei tarvitse lisävirtaa multiprogrammerilta toimiakseen vaan saa käyttöjännitteensä sarjaporttiliitännän kautta. Led-kortteja on mahdollista ketjuttaa, ja kortin ledit on nimetty D0-D7-ledeiksi.

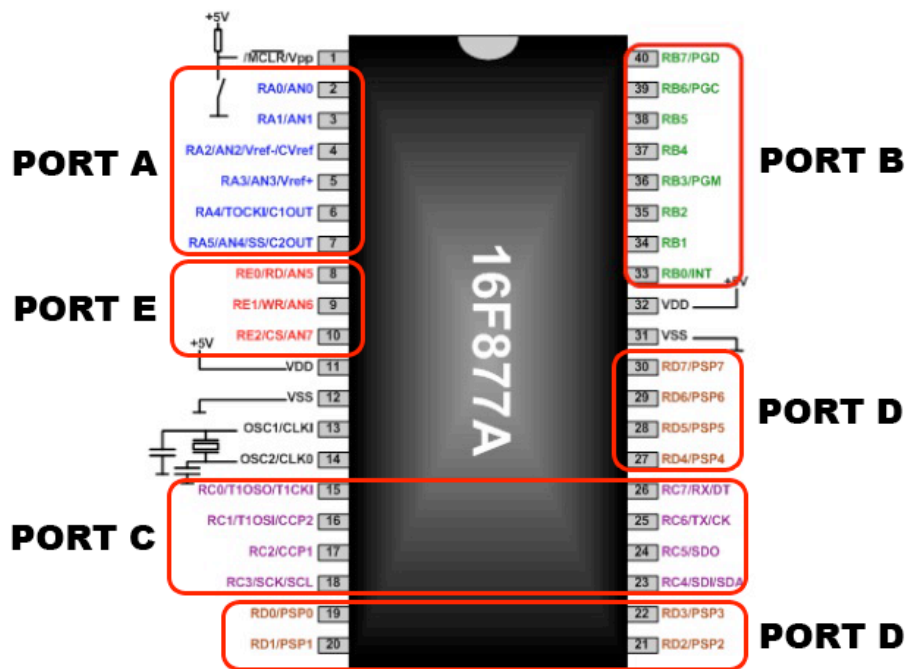
Led-kortti toimii samalla periaatteella kuin painonappikortti; ledit kuvaavat kahdeksaa bittä ja niitä ohjataan syöttämällä led-kortin porttiin tavumuotoinen luku 0-255 väliltä. Luku 255 tarkoittaa binaärilukua 1111111, jolloin kaikki ledit syttyvät. Vastaavasti led-kortille lähetetty luku 0 tarkoittaa ettei yksikään ledi pala rivistössä.

2.2 PIC-16F788A prosessori

E-Blocks -sarjan multiprogrammer tukee PICmikron sekä AVR-mikrokontrollereita. Mikrokontrollerin valinta ja sen ominaisuudet vaikuttavat siihen, millaisia laitteita ja kuinka laajoja ohjelmistoja mikrokontrolleri kykenee hallitsemaan. Tutkintotyön laitteistoissa käytettiin PIC-16F788A mikrokontrollereja ja seuraavassa on esitelty tarkemmin mikrokontrollerin teknisiä ominaisuuksia.

PIC-16F788A on 40-nastainen PIC-mikrokontrolleri, jossa on 8000-ohjelmointikäskyn flash-muisti, 368 tavua RAM-muistia sekä 256 tavua EEPROM-muistia. /14/ Työssä käytettiin mikrokontrollerin sisäistä kidettä (multiprogrammerilta kytkin XTAL-asentoon), jonka nopeus oli 19 660 800Hz.

E-Blocks -sarjan multiprogrammer mahdollistaa oheislaitteiden helpon liittämisen mikrokontrollerin nastoihin. Seuraavassa kuvassa havainnollistuu kuinka eri multiprogrammerin RS323-portit ovat yhteydessä käytettyyn mikrokontrolleriin (kuva 12).



Kuva 12. 16F877A mikrokontrollerin nastat sekä multiprogrammerin vastaavat portit /15/

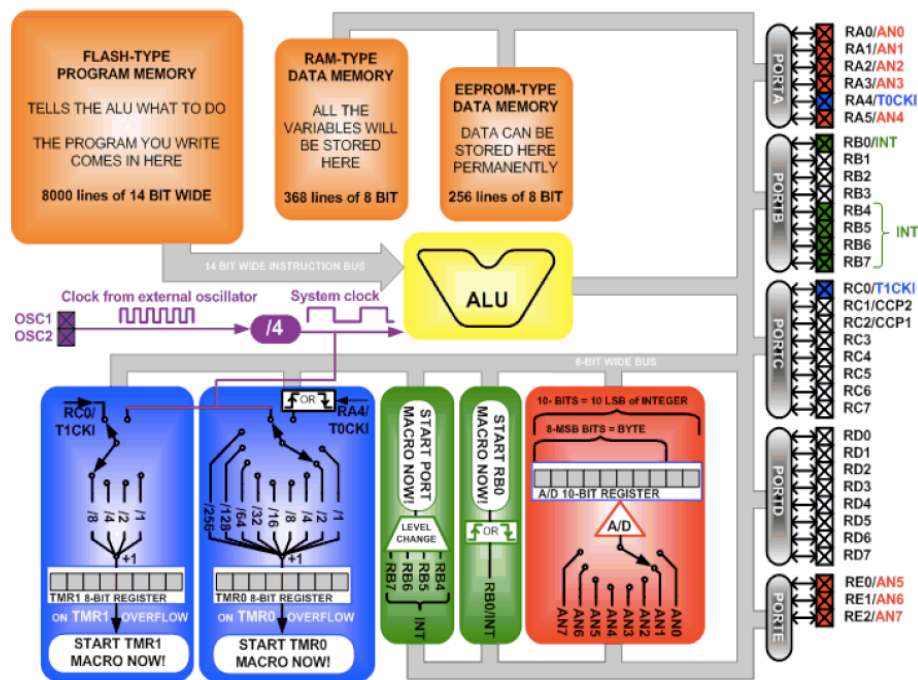
Portti A on kuusinastainen liitäntä, ja liitäntää voidaan käyttää digitaalisena sekä analogisena sisäänmenoina. Portissa on viisi analogista sisäänmenoa ja portti mahdollistaa myös digitaalisen ulostulon.

Portti B käyttää kahdeksaa nastaa ja porttia voidaan käyttää digitaalisina sisäänmenoina sekä ulostuloina. RB0 eli nastaa 33 voidaan käyttää ulkoisen keskeytyksen nastana sekä RB4-RB7 toimivat tarvittaessa ulkoisena porttikeskeytyksenä.

Portti C käyttää kahdeksaa nastaa ja toimii digitaalisina sisäänmenoina sekä ulostuloina. Portti C on myös ainoa portti, joka sisältää USARTin (Universal Synchronous – Asynchronous Receiver Transmitter) sarjaliikennettä varten. ZigBee-kortit tarvitsevat USART-ominaisuutta, joten työssä käytetyt ZigBee-kortit liitetään aina multiprogrammerin porttiin C.

Portti D käyttää myös kahdeksaa nastaa ja sitä voidaan käyttää digitaalisina sisäänmenoina että ulostuloina.

Portti E käyttää vain kolmea nastaa ja sitä voidaan käyttää digitaalisina sisäänmenoina sekä ulostuloina. Portissa on myös kolme analogista sisäänmenoa.



Kuva 13. PIC-16F788A mikrokontrollerin yksinkertaistettu lohkoakaavio /16/

Mikrokontrollerin portit on kytketty mikrokontrollerin muihin sisäisiin piireihin 8-bittisellä väylällä. Kuvassa 13 on havainnollistettu tarkemmin 16F788A mikrokontrollerin sisäiset lohkot.

Kuvan 13 yläosassa sijaitsevat Flash-, RAM-, sekä EEPROM-muistilohkot. Mikrokontrollerin ohjelma sijaitsee Flash-muistissa, ohjelman muuttujat tallennetaan RAM-muistilohkoon ja EEPROM-muistilohkoon voidaan tallentaa pysyvää tietoa, kuten valmiita taulukoita.

Kuvan keskellä sijaitseva ALU, Aritmeettis-looginen yksikkö, on PICmikron sydän, jonka läpi kaikki mikrokontrollerin tieto kulkee. Mikrokontrollerin ohjelma sijaitsee flash-muistissa, josta ohjelma käskyyttää ALU:a. ALU on yhteydessä PICmikron portteihin sekä lohkoihin 8-bittisen dataväylän kautta.

Kaavion vasemmassa alakulmassa sijaitsee järjestelmäkello TMR1 sekä TMR2 eli Timer1 ja Timer2, joiden tehtävänä on tuottaa PICmikrolle tarkkaa tietoa ajasta. Järjestelmäkello toimii 4 kertaa hitaammin kuin ulkoinen oskillaattorikello.

PICmikron A/D-muunnin, eli analogia-digitaalimuunnin sijaitsee oikealla alhaalla ja mikrokontrollerin portit A-E on esitelty kuvaajan oikeassa reunassa.

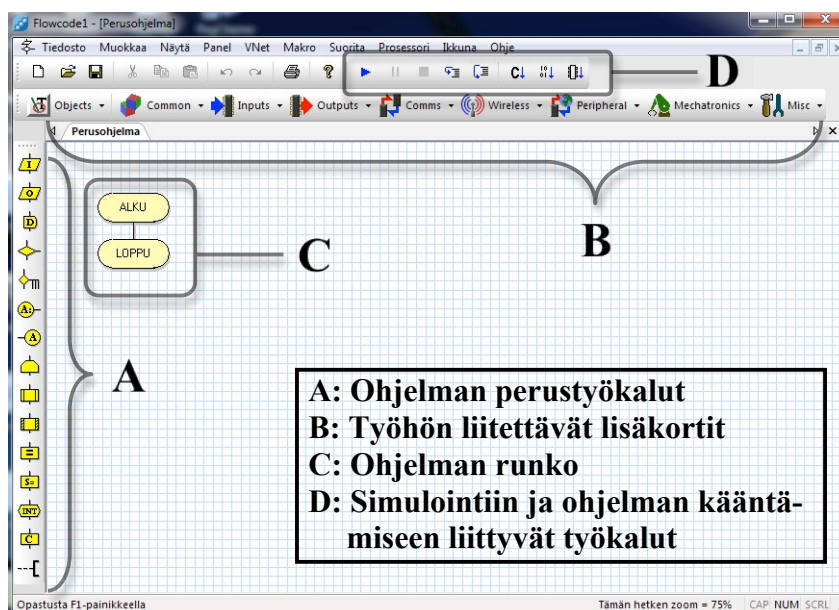
3 FLOWCODE

Flowcode on Matrix Multimedian kehittämä mikrokontrollerien helposti omaksuttavaan ohjelmointiin erikoistunut ohjelma, joka on ehtinyt jo neljänteen versioonsa. Ohjelmassa valitaan käytettävä mikrokontrolleri ja siihen liitetyt laitteet, jonka jälkeen itse mikrokontrollerin ohjelma ohjelmoidaan yksinkertaisella vuokaavio-ohjelmoinnilla. Ohjelmaa voidaan tarvittaessa simuloida tai siirtää ohjelma mikrokontrollerille USB-johdon avulla. Flowcode kääntää ohjelmoidun vuokaavion ensin C-kielelle, sen jälkeen assembly-kielelle, josta ohjelma käännetään mikrokontrollerin ymmärtämäksi heksakoodiksi. Lopulta ohjelma voidaan siirtää USB-liitännällä mikrokontrollerin muistiin.

Flowcode on ohjelmoitu niin, että käyttäjän ei välttämättä tarvitse ymmärtää laajasti mikrokontrollerien arkkitehtuuria tai osata kehittyneitä ohjelmointia. Flowcoden avulla käyttäjä voi keskittyä suunnittelemaan ja toteuttamaan laitteiston haluttua toimintalogiikkaa. Riittää, että ohjelmalle kertoo käytetyn mikrokontrollerin sekä mihin multiprogrammerin porttiin on mikäkin lisäkortti liitetty.

3.1 Flowcoden työkalut

Flowcoden perusnäkökulma on yksinkertainen. Kuvassa 14 on esitelty Flowcoden perusnäkökulma sekä päävalikot.



Kuva 14. Flowcoden perusnäkökulma sekä valikot

Flowcoden perusnäkyssä kuvassa 14 näkyvä A-valikko sisältää vuokaavio-ohjelmoinnin perustyökalut, kuten sisääntulojen sekä ulostulojen määrittelyt, jos-lauseet, viiveet, silmukat ja niin edelleen. Valikosta kutsutaan myös käyttäjän itse ohjelmoimat makrot, sekä esimerkiksi LCD-näytön ohjaukseen tai ZigBee-kortin toimintoihin liittyvät valmiit komponenttimakrot.

Kuvan 14 B-valikosta valitaan multiprogrammeriin liitetyt lisäkortit. Varsinainen ohjelma ohjelmoidaan vetämällä A-valikosta haluttuja toimintoja ohjelman runkoon alkupisteen ja loppupisteen väliin (C-alue). Tehdyn ohjelman simulointi ja ohjelman mikrokontrollerille lähetys onnistuu valikon D-painikkeista.

3.2 Esimerkkiharjoitukset

Seuraavassa tutustutaan Flowcode-ohjelmoinnin keskeisiin perusasioihin, sekä vuokaavio-ohjelmoinnin logiikkaan. Esimerkkiharjoituksissa on tarkoitus esitellä miten Flowcodella ohjelmoidaan yksinkertaisia ohjelmia, sekä kuinka työssä käytettyjen ZigBee-korttien välille muodostetaan yksinkertainen verkko. Esimerkkiharjoituksissa on käytetty Flowcode Pro v4.2.3.58 ohjelmistoa.

3.2.1 Ennakkovalmistelut

Ensimmäisenä työvaiheena on päättää millaisia lisäkortteja ja mikroprosessoria työssä on tarkoitus käyttää. Harjoituksessa käytetty mikroprosessori on PIC-16F778A, joten päätettäväksi jää muiden korttien valinta. Kun kortit on valittu, kiinnitetään ne huolellisesti multiprogrammerin RS323-portteihin ja varmistetaan, että lisävirtaa tarvitsevat kortit saavat tarvitsemansa virran multiprogrammerilta. Kun korttien kytkennät on tehty, voidaan multiprogrammeriin kytkeä virta.

Aluksi Flowcode-ohjelmasta valitaan uusi tiedosto, jonka jälkeen valitaan käytettävä mikrokontrolleri. Kun Flowcoden päänäky käynnistyy, on tarpeen varmistaa, että prosessorivalikon kohdasta ”Konfiguroi” oskillaattoriksi on valittu ”XTAL”. Tämän jälkeen valitaan multiprogrammeriin liitetyt lisäkortit kuvan 14 C-valikosta ja työhön liitettyjen korttien asetuksista kerrotaan ohjelmalle mikä kortti on kytketty mihinkin porttiin. Korttien asetuksiin pääsee klikkaamalla avattujen korttien virtuaalisia kuvakkeita hiiren oikealla painikkeella ja valitsemalla ”Connections” sekä tarvittaessa myös ”Ext properties”.

3.2.2 LCD-näytölle kirjoittaminen

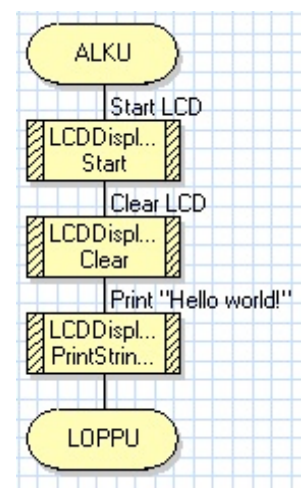
Ensimmäisessä esimerkkiharjoituksessa esitellään, kuinka Flowcodella saa kirjoitettua yksinkertaisen lauseen kaksiriviselle mustavalkoiselle LCD-näytölle. Työssä käytetty LCD-näyttökortti on kytketty multiprogrammerin porttiin C ja Flowcodessa on määritetty LCD-kortin ominaisuudet vastaamaan kytkentää.

LCD-näytön ohjelmoinnissa käytetään hyväksi Flowcoden valmiita komponenttimakroja, jotka ilmaantuvat valittaviksi kun LCD-näyttö liitetään Flowcodeen kuvan 14 B-valikosta. Komponenttimakrot ovat C-kielellä valmiiksi ohjelmoituja toimintoja kuten LCD-näytön alustus, näytön tyhjennys tai esimerkiksi sanan kirjoitus näytölle. Normaalitapauksissa käyttäjän ei tarvitse käydä muuttamassa valmiita komponenttimakroja.

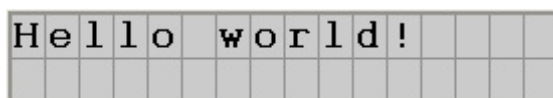
Esimerkin toteutus alkaa tekemällä aiemmin esitetty ennakkovalmistelut. Ensimmäiseksi ohjelman runkoon vedetään vasemmasta toimintopalkista uusi komponenttimakro, valitaan makron asetuksista "LCDDisplay(0)" ja makrokseksi "Start". Toiminto alustaa LCD-ruudun ohjelman käyttöön. Tämän jälkeen vedetään uusi komponenttimakro edellisen alapuolelle ja valitaan makron asetuksista "LCDDisplay(0)" sekä makrokseksi "Clear".

Tämän jälkeen raahataan edellisten alapuolelle vielä kolmas komponenttimakro ja valitaan makron asetuksista "LCDDisplay(0)", makrokseksi "PrintString" ja parametriksi "Hello world!" mukaan lukien lainausmerkit.

Ohjelman rungossa on nyt allekkain kolme komponenttimakroa (kuva 15), joista ensimmäinen alustaa LCD-ruudun, toinen tyhjentää LCD-ruudun sekä kolmas kirjoittaa ruudulle lauseen "Hello world!".



Kuva 15. LCD-ohjelma



Kuva 16. Simulaation tulos

Ohjelman voi simuloida (kuva 16) ja mikäli simulointi toimii oikein, se voidaan siirtää mikrokontrollerille kytkemällä USB-johto tietokoneen sekä multiprogrammerin väliin ja valitsemalla kuvan 14 D-valikosta ”Käännä prosessorille” painike. Tällöin Flowcode kääntää ohjelman mikrokontrollerin ymmärtämään muotoon ja siirtää USB-johtoa pitkin ohjelman mikrokontrollerin muistiin.

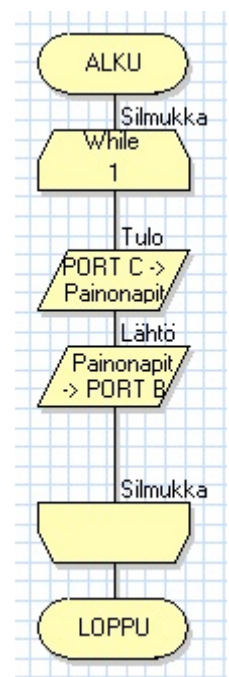
3.2.3 Ledien sytytys painonapeista

Tässä esimerkissä yhteen multiprogrammeriin kytketään kaksi korttia, led-kortti sekä painonappikortti. Tarkoituksena on ohjelmoida yksinkertainen ohjelma, jossa ohjelma tutkii mitä painonappeja painetaan, ja lähettää sytyttää nappeja vastaavat led-valot.

Työssä käytetty led-kortti sisältää kahdeksan punaista lediä, jotka kuvaavat kahdeksaa bittä. Ledejä voidaan ohjata helposti lähettämällä led-kortin porttiin numero tavumuodossa. Esimerkiksi numero 1 tarkoittaa binäärilukua 00000001, jolloin ensimmäinen led syttyy. Mikäli halutaan sytyttää kaikki kahdeksan led-valoa, tulee porttiin lähettää numero 255, mikä vastaa binäärilukua 11111111. Vastaavasti luku 0 sammuttaa kaikki ledit. Työssä led-kortti kytketään porttiin B.

Painonappikortissa on kahdeksan painonappia ja kortti kytketään porttiin C. Painonapit vastaavat 8-bitin sarjaa ja painonappien tila näkyy mikrokontrollerille vastaavana 8-bitin tavuna, numero 1 vastaa pohjaan painettua nappia ja numero 0 vapaana olevaa nappia.

Ohjelmointi aloitetaan raahaamalla kuvan 14 A-valikosta ohjelman runkoon uusi silmukka, jossa ohjelma pyörii niin kauan, kunnes laitteesta sammutetaan virrat. Silmukan arvo on oletuksena 1, jolloin ohjelma pyörii silmukassa loputtomasti. Silmukan sisälle raahataan uusi sisään-tulo-laatikko ja lisätään uusi muuttuja ”Muuttujat”-painikkeesta. Muuttujan nimeksi asetetaan ”Painonapit” ja muuttujan tyyppi ”byte” eli tavu. Tämän jälkeen palataan takaisin tulon asetuksiin ja valitaan muuttujaksi juuri tehty muuttuja ”Painonapit” ja valitaan portiksi C. Tämä tarkoittaa sitä, että ohjelma tallentaa portista C luetun arvon tavu-tyyppiseen muuttujaan ”Painonapit”.



Kuva 17.
Valmis ohjelma

Tämän jälkeen raahataan silmukan sisälle tulon alapuolelle uusi lähtö-laatikko ja avataan lähdön ominaisuudet. Muuttujaksi valitaan ”Painonapit” ja portiksi B (kuva 17). Tällöin muuttujan ”Painonapit” sisältämä arvo lähetetään porttiin B.

Ohjelma on valmis ja sitä voidaan simuloida valitsemalla simuloinnin käynnistyksen painike kuvan 14 D-valikosta. Käytännössä ohjelma kierrättää ikuista silmukkaa, jossa ohjelma ensin tarkistaa painonappien tilan, tallentaa tilan muuttujaan ”Painonapit” ja lähettää muuttujan arvon ledeille. Mikäli simulointi toimii oikein, ohjelma voidaan siirtää mikrokontrollerille USB-johdolla kuvan 14 D-osion ”Käännä prosessorille” painikkeesta.

3.3 Zigbee-verkon luominen ja hyödyntäminen

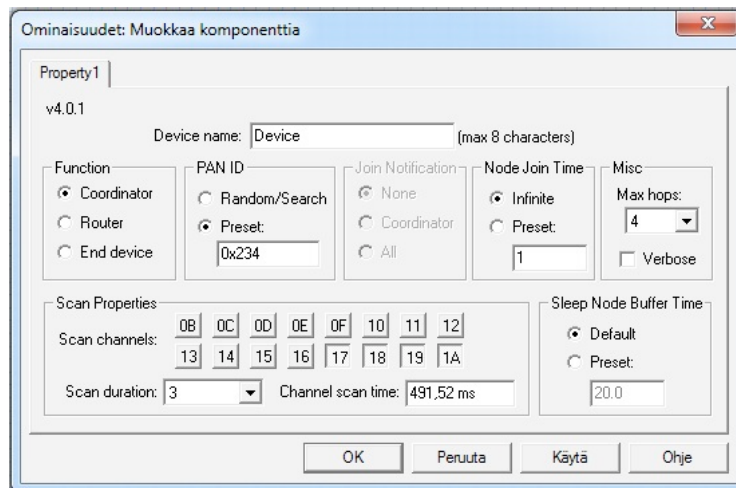
E-Blocks -sarjaan on saatavissa kahdenlaisia ZigBee-kortteja, koordinaattoreita sekä reititin- / päätelaitekortteja. ZigBee-kortit tarvitsevat toimiakseen mikrokontrollerin USART-ominaisuutta, joten ZigBee-kortit tulee kytkeä multiprogrammerin porttiin C.

Seuraavassa on esitelty kuinka Flowcodella ohjelmoidaan yksinkertainen verkko kahden laitteen välille ZigBeellä. Esimerkissä ohjelmoidaan edellisen harjoituksen kaltainen ledien sytytys painonapeista, mutta tällä kertaa painonapit ja led-kortti sijaitsevat eri multiprogrammereissa. Painonappien arvo lähetetään päätelaitteelta koordinaattorille ZigBeen välityksellä.

3.3.1 Koordinaattori

Esimerkkiharjoituksessa multiprogrammerin porttiin B kytketään 8-ledin kortti ja porttiin C kytketään ZigBee koordinaattori-kortti. Tämän jälkeen tehdään aiempien esimerkkien mukaiset Flowcoden ennakkovalmistelut.

ZigBee-kortti lisätään Flowcodeen valitsemalla kuvan 14 mukaisen B-palkin ”Wireless” -valikosta ZigBee, jonka jälkeen ZigBeen logo ilmantuu näkyviin laitepaneeliin. ZigBee-kuvakkeen ”Ext properities”-valikosta käytettävän kortin asetukset. Kuvassa 18 on esitelty ZigBee-korttien asetukset sekä selitetty tarkemmin asetusten erilaiset valintavaihtoehdot .



Kuva 18. ZigBee-kortin asetukset

Device name	ZigBee-kortin nimi verkossa.
Function	Valitaan toimiiko käytettävä kortti koordinaattorina, reitittimenä vai päätelaitteena.
PAN ID	Yksilöllinen verkon tunniste, jonka avulla verkon laitteet osaavat liittyä samaan ZigBee-verkkoon.
Join notification	Mikäli kortin tehtävänä on toimia reitittimenä tai päätelaitteena, tästä valitaan kenelle kortti ilmoittaa liittyessään verkkoon.
Node Join Time	Valitaan koordinaattorille aika, kauanko verkkoon saa liittyä muita laitteita.
Misc	Valitaan maksimimäärä toistettaville broadcast-lähetyksille ja voidaan kytkeä päälle LCD-näytön tarkan lokin keräys.
Scan Properties	Valitaan kanavat joilta laitteet skannaavat verkkoa sekä aika kuinka kauan kanavia skannataan.
Sleep Node Buffer Time	Valitaan laitteen virransäästöön liittyvät aikamääritykset.

Koordinaattorille määritetään seuraavat asetukset:

- Function: Coordinator
- Device name: Coord
- PAN ID: preset
- Join Notification: None
- Node Join Time: Infinite
- Misc: Max Hops = 4; Verbose = tyhjä
- Scan Channels: 17, 18, 19, 1A; Scan Duration = 3; Sleep = Default

Ensimmäiseksi ohjelmarunkoon raahataan uusi laskutoimitus, lisätään uusi byte-tyyppinen muuttuja ”input” ja alustetaan muuttuja arvoksi 0. Tämän jälkeen ohjelmarunkoon lisätään uusi komponenttimakro, jossa komponentin ”Zigbee(0)” makroksi valitaan ”Init_Network”. Tämän jälkeen lisätään toinen komponenttimakro, jonka komponentin ZigBee(0) makroksi valitaan ”Connect_To_Name” ja parametriksi ”End1”.

Koordinaattorin ohjelma siis alkaa alustamalla ensin input-muuttuja arvoksi 0, sitten käynnistää ZigBee-verkon ja lopuksi yhdistää päätelaitteeseen, jonka nimi on ”End1”.

Seuraavaksi ohjelmaan lisätään silmukka, jonka sisälle lisätään uusi komponenttimakro. Komponentin Zigbee(0) makroksi valitaan ”Receive_Char”, nTimeout parametrin arvoksi 100 ja paluuarvoksi muuttuja input. Viimeiseksi silmukkaan lisätään ulostulo, jonka asetuksiin määritetään muuttujan input arvon lähetyksen porttiin B.

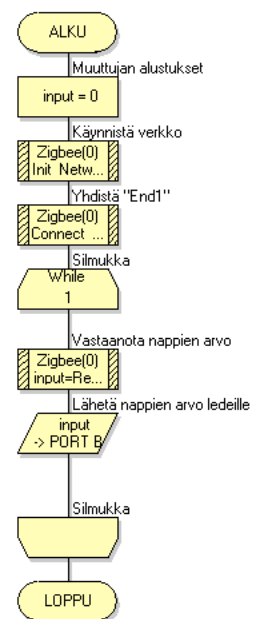
Silmukassa koordinaattori siis kuuntelee jatkuvasti tulevaa lähetystä, tallentaa saapuvan lähetyksen arvon muuttuun input ja lähettää tallennetun arvon led-kortille.

Esimerkkiharjoituksen koordinaattorin valmis vuokaavio on esitetty kuvassa 19.

3.3.2 Päätelaite

Esimerkissä päätelaitteen multiprogrammerin porttiin B kytketään painonappikortti ja porttiin C ZigBee päätelaite-kortti. Tämän jälkeen painonappikortille kytketään kortin tarvitsema lisävirta multiprogrammerilta ja tehdään aiempien esimerkkien mukaiset Flowcoden ennakkovalmistelut. Flowcoden päänäkymässä lisätään projektiin liitetyt kortit ja valitaan asetuksista laitteiden portit vastaamaan multiprogrammeriin liitetyjä kortteja.

Päätelaitteen Flowcoden ZigBee-asetuksina käytetään samoja kuin koordinaattorin, mutta ”Function” arvoksi asetetaan ”End device” ja nimeksi ”End1”



Kuva 19.
Koordinaattori

Päätelaitten ohjelmointi aloitetaan samalla tavalla kuin koordinaattorin, ensimmäiseksi tehdään uusi byte-tyyppinen muuttuja ”input” ja alustetaan muuttuja arvoksi 0. Tämän jälkeen ohjelmarunkoon lisätään uusi komponenttimakro, jossa komponentin ”Zigbee(0)” makrokseksi valitaan ”Init_Network”. Tämän jälkeen lisätään toinen komponenttimakro, jonka komponentin ZigBee(0) makrokseksi valitaan ”Connect_To_Coordinator”.

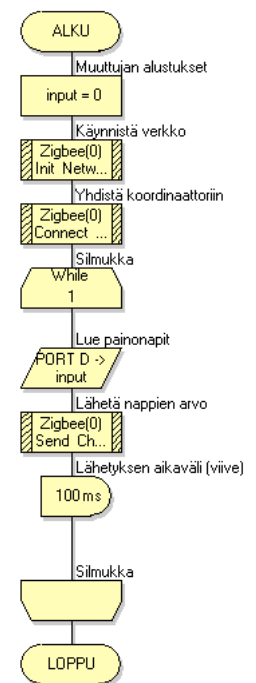
Päätelaitteen ohjelma siis alustaa ensimmäisenä input-muuttujan arvoksi 0, sitten käynnistää ZigBee-verkon ja lopuksi yhdistää koordinaattoriin.

Ohjelman runkoon lisätään seuraavaksi silmukka, jonka sisällä varsinainen ohjelma verkon käynnistymisen jälkeen toimii. Silmukan sisälle lisätään sisääntulo, jonka asetuksissa määritetään D-portista mitatun arvon tallentuvaksi muuttujaan input. Sisääntulon alapuolelle lisätään uusi komponenttimakro, jonka komponentin Zigbee(0) makrokseksi valitaan Send_Char ja parametriksi muuttuja input. Tämän jälkeen silmukan viimeiseksi toiminnoksi lisätään 100ms viive.

Ohjelma siis lukee silmukassa painonappien tilan, tallentaa tilan arvon muuttujaan input ja lähettää muuttujan arvon ZigBeellä ilmarajapintaan. Määritetyn viiveen jälkeen ohjelma lukee uudelleen painonappien tilat ja lähettää tilan arvon jälleen eteenpäin.

Silmukan viimeisenä toimintona olevan viiveen avulla voidaan määrittää kuinka nopealla aikavälillä painonappien uusi arvo luetaan ja lähetetään eteenpäin. Lyhyt arvo mahdollistaa painonappien lyhyen lukemisviiveen, mutta toisaalta myös lisää ilmassa lähetettävien pakettien määrää. Asettamalla viive 100 millisekunnin sijaan esimerkiksi yhdeksi sekunniksi, koordinaattori lähettää painonappien tilan eteenpäin vain kerran sekunnissa. Tämä vähentää huomattavasti lähetettävien pakettien määrää, mutta myös lisää viivettä kuinka nopeasti koordinaattori reagoi painonappien tilan muutoksiin.

Esimerkkiharjoituksen päätelaitteen valmis vuokaavio on esitetty kuvassa 20.



Kuva 20.
Päätelaite

4 ZIGBEE-JÄRJESTELMÄN SUUNNITTELU, TOTEUTUS JA ANALYSOINTI

Matrix Multimedian E-Blocks -sarja mahdollistaa myös vaativampien järjestelmien suunnittelun ja toteutuksen. Tässä tutkintotyössä suunniteltiin ja toteutettiin viiden erillisen laitteen muodostama ZigBee-verkko, jossa neljä päätelaitetta olivat yhteydessä koordinaattoriin. Seuraavassa on esitelty tarkemmin ZigBeen teknologiaa ja sen tarjoamia mahdollisuuksia sekä työn eri vaiheet ja toteutustavat.

4.1 Miksi ZigBee?

Ensimmäinen ZigBee-standardi julkaistiin ensimmäisen kerran joulukuussa 2004. ZigBee tarkoittaa lyhyen kantaman tietoliikenneverkkoa, joka noudattaa IEEE 802.15.4-standardia ja toimii 868MHz, 915MHz tai 2,4GHz taajuudella. /17/ ZigBeen tiedonsiirtonopeus on parhaimmillaankin vain 250kb/s, joten se soveltuu parhaiten esimerkiksi anturitietojen lähetykseen.

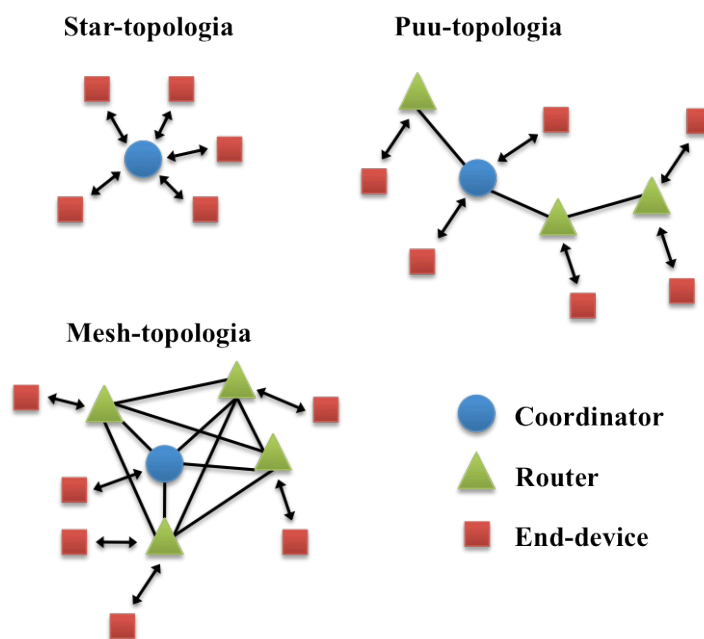
868MHz sekä 915MHz taajuudet käyttävät BPSK modulaatiota ja 2,4GHz taajuudella käytössä on OQPSK modulaatio. ZigBee-standardi käyttää hyväkseen CSMA/CA törmäyksenestoa, jolloin lähettäjä kuuntelee radiorajapintaa ja lähettää vasta kun kukaan muu ei lähetä tietoa.

ZigBee mahdollistaa jopa 256 laitteen verkkokokonaisuuksia ja standardi tukee tehokasta 128-bittistä AES salausalgoritmia. Yksi suurimmista ZigBeen vahvuuksista on päätelaiteiden tehokkaat virransäästötilat, joiden avulla esimerkiksi lämpötilatietoa lähettävä anturi saattaa toimia jopa 2 vuotta yhdellä paristolla. ZigBee-päätelaite saattaa siis olla unitilassa jopa 99,99% ajastaan. Standardi mahdollistaa päätelaitteen unitilasta heräämisen alle 15 millisekunnissa, liittymisen verkkoon alle 30 millisekunnissa, ja lähetyksen aloittamisen alle 15 millisekunnissa /17/. Päätelaite voidaan asettaa esimerkiksi heräämään unestaan kymmenen minuutin välein, mittaamaan lämpötilan ja lähettämään tiedon koordinaattorille. Tämän jälkeen päätelaite voi vaipua takaisin uneen, jolloin päätelaitteen tehontarve on alimmillaan vain 1µA luokkaa. /18/

ZigBeen ensisijaisia käyttökohteita ovat teollisuuden sekä kotiautomaation tarpeet. ZigBeen avulla voidaan muodostaa turvallisia ja vähän virtaa tarvitsevia anturiverkkoja, joi-

den avulla voidaan mitata esimerkiksi tehtaan lämpötilatietoja, kerätä paineantureilta tietoa, ohjata moottoreita tai valoja ja niin edelleen.

ZigBee-standardi mahdollistaa kolmen erilaisen verkkotopologian käytön. ZigBee-verkko laitteiden välille voidaan muodostaa käyttämällä tähtitopologiaa, puutopologiaa tai mesh-topologiaa. /19/



Kuva 21. ZigBee-verkkojen erilaiset topologiat

Star-topologiaa, eli tähti-topologiaa, käyttävässä ZigBee verkossa kaikki päätelaitteet ovat yhteydessä suoraan koordinaattoriin.

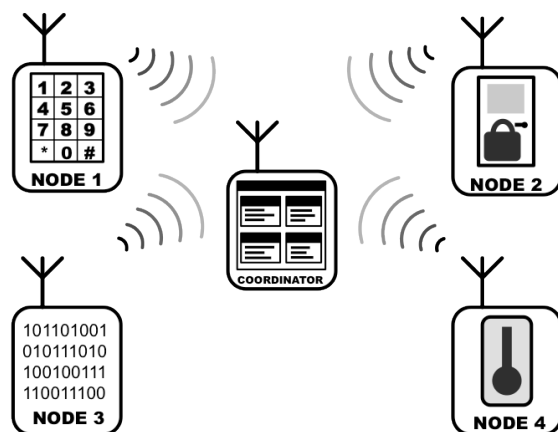
Puu-topologiaa hyväksikäyttävässä verkossa hyödynnetään reitittimiä verkon kantaman laajentamiseksi. Päätelaitteet yhdistävät joko reitittimeen tai suoraan koordinaattoriin ja reitittimet välittävät verkossa liikkuvat paketit koordinaattorin sekä päätelaitteiden välillä.

Mesh-topologian verkot ovat älykkäimpiä ZigBee-verkkoja ja kykenevät dynaamiseen reititykseen. Reitittimet ovat yhteydessä sekä koordinaattoriin että toisiinsa ja osaavat reitittää verkossa siirrettävät paketit lyhyintä mahdollista reittiä. Mesh-verkkoja käyttämällä verkon viiveet vähenevät sekä verkon luotettavuus kasvaa älykkään reitityksen ansiosta. Vaikka jokin reitittimistä lakkaisi toimimasta, verkon reititys mukautuu tilanteen mukaan.

ja verkko jatkaa toimintaansa. Päätelaitteet osaavat myös yhdistää itsenäisesti lähimpään reitittimeen tai koordinaattoriin.

4.2 Järjestelmän suunnittelu

Järjestelmän suunnittelun alkuvaiheissa päätettiin tehdä star-topologiaa hyväksikäyttävä ZigBee-verkko, jossa yksi koordinaattori hallitsisi kolmea eri toimintakokonaisuutta (kuva 22). Ensimmäinen toiminto olisi oven etäavaus salasanan avulla. Ensimmäinen päätelaite huolehtisi nelinumeroisesta tunnusluvusta, jonka käyttäjä painelee numeronäppäimistölle, ja toinen päätelaite ohjaisi oven moottoria. Mikäli käyttäjä näppäilisi oikean salasanan, ilmoittaisi ensimmäinen päätelaite asiasta koordinaattorille ja koordinaattori käskyttäisi oven moottoria aukeamaan. Toinen toiminto olisi 8-bitin binääriluvun lähetyks kolmannelta päätelaitteelta koordinaattorille ja kolmas toiminto olisi lämpötilatiedon lähetyks neljänneltä päätelaitteelta koordinaattorille.



Kuva 22. Työssä käytetyt star-topologian päätelaitteet

4.3 Käytettävän laitteiston mahdollisuudet ja rajoitukset

Tässä työssä suurimmiksi haasteiksi nousivat verkon liikenteen sekä lähetettävien pakettien toimintalogiikan suunnittelu, sekä käytetyn mikrokontrollerin sisäisen muistin koko. Alusta asti oli selvää, että koordinaattorin mikroprosessorin hallittavaksi tulisi laitteiston laajin ohjelmisto. Yhtenä huolenaiheena olikin pitää silmällä, että ohjelmisto mahtuisi kokonaisuudessaan koordinaattorin mikrokontrollerille. Koordinaattorin ohjelma ohjelmoitiin toiminto kerrallaan ja ohjelmiston kokoa tarkkailtiin. Lopullinen toteutettu ohjelmisto mahtui kuitenkin hyvin käytettyyn mikroprosessoriin eikä ongelmia tässä suhteessa kuitenkaan syntynyt.

4.4 Järjestelmän toteutus

Laitteiston suunnittelu aloitettiin määrittelemällä laitteiston halutut toiminnot ja ominaisuudet. Kun laitteiston toiminnot oltiin päätetty, suunniteltiin sen jälkeen laitteiden väliset pakettien tiedonsiirtologiikat. Vaikka ZigBee on valmis tiedonsiirtostandardi, työssä toteutettiin tarkoituksella myös oma laitteiden välinen pakettien lähetysoogiikka.

Koordinaattorin sekä päätelaitteiden välinen tiedon lähetyso ja vastaanotto toteutettiin seuraavalla tavalla; aina kun päätelaite lähetti koordinaattorille tietoa, lähetettiin jokaisella lähetysoerralla peräkkäin kolme tavua. Ensimmäinen lähetettävä tavu sisälsi päätelaitteen node-numeron, jotta koordinaattori tunnisti kuka tietoa yritti lähettää. Sen jälkeen päätelaite lähetti varsinaisen informaatiotavunsa ja lopuksi päätelaite lähetti päätelaitteen node-numerosta sekä lähetettävästä informaatiotavusta yhteenlasketun pariteettitavun.

Koordinaattori kuunteli jatkuvasti lähettikö kukaan uutta tietoa. Mikäli koordinaattori kuuli lähetysoen, tarkisti se ensin ensin mikä päätelaite lähettää, ja sen jälkeen vastaanotti lähettäjän informaatiotavun ja lopuksi pariteettitavun. Tämän jälkeen koordinaattori vertasi täsmäkö vastaanotetun lähettäjän node-numerosta sekä informaatiotavusta yhteenlaskettu pariteetti vastaanotettua pariteettia. Mikäli pariteetti täsmäsi, koordinaattori hyödynsi vastaanottamaansa uuden informaation ja esimerkiksi päivitti näytöllensä uuden lämpötilatiedon.

Päätelaitteiden koordinaattorille lähettämä informaatiotulva minimoitiin ohjelmoimalla päätelaitteisiin logiikka, joka tutki muuttuiko päätelaitteen lähettämä informaatio vai pysyikö se samana. Jos päätelaitteen informaatio ei muuttunut, tieto lähetettiin koordinaattorille vain noin viiden sekunnin välein uudelleen. Jos päätelaitteen informaatio muuttui, uusi informaatio, kuten muuttunut lämpötilatieto, lähetettiin koordinaattorille välittömästi.

Koska päätelaitteet lähettivät koordinaattorille tietonsa joka tapauksessa noin viiden sekunnin välein, voitiin koordinaattoriin ohjelmoida toiminto, joka mittasi päätelaitteiden signaalinvoimakkuudet aina kun päätelaitteelta vastaanotettiin uusi lähetyso. Päätelaitteiden signaalinvoimakkuudet myös tulostettiin näytölle. Mikäli lähetysoä ei saatu noin 15 sekunnin kuluessa, merkittiin päätelaite poissaolevaksi "N/A"-tunnuksella.

Seuraavassa on esitelty koordinaattorin sekä päätelaitteiden tarkemmat toteutukset.

4.4.1 Koordinaattorin toteutus

Koordinaattorin tehtävänä oli käynnistää ZigBee-verkko, sekä vastaanottaa ja hyödyntää päätelaitteiden lähettämää informaatiota. Työssä käytetyssä koordinaattori-yksikössä käytettiin graafista LCD-näyttöä, jolle tulostettiin päätelaitteiden lähettämät informaatiot. Graafisen LCD-näytön avulla voitiin havainnollisesti piirtää näytölle huomattavasti suurempi määrä tietoa verrattuna kaksiriviseen mustavalkoiseen LCD-näyttöön. Koordinaattorin vastaanottama informaatio näytettiin näytöllä kuvan 23 mukaisesti.

Coordinator V0.76	
Sig. lvl	Security
N1: N/A %	Door open
N2: 78 %	Correct Password
N3: 81 %	
N4: 65 %	
N3 byte	Temperature
00100011	Now: 103
35	Min: 32
0x23	Max: 189

Kuva 23. Koordinaattorin graafisen näytön ulkoasu

”Sig. lvl” – ikkunaan tulostettiin kaikkien neljän päätelaitteiden signaalivoimakkuudet, ja mikäli koordinaattori ei saanut päätelaitteelta lähetystä noin 15 sekunnin kuluessa, laite merkattiin poissaolevaksi ”N/A” tunnuksella.

”Security”-ikkunaan tulostettiin tieto siitä, oliko ensimmäiseen päätelaitteeseen syötetty salasana oikein vai väärin sekä oliko toisen päätelaitteen ohjaaman oven tila auki vai kiinni. Kolmannen päätelaitteen lähettämä tavutieto tulostettiin ”N3 byte”-ikkunaan ensin binaärilukuna, sitten kokonaislukuna ja lopulta heksalukuna.

Päätelaitteen 4 lähettämä lämpötilatieto näytettiin ”Temperature”-ikkunassa. Ikkunaan tulostettiin nykyinen lämpötila sekä lämpötilan minimi- että maksimi-arvot. Lämpötilatieto vastaanotettiin 8-bitin tavumuodossa, joten lämpötila-arvot vaihtelevat 0-255 väliltä. Tavumuotoisen lämpötilatiedon muuttaminen celsius-asteikolle jätettiin toteuttamatta, sillä operaatio ei olisi ollut työssä tarkasteltavan ZigBee-verkon luomisen ja toteutuksen kannalta merkityksellinen.

Koordinaattorin täydellinen vuokaavio on liitteenä 2.

4.4.2 Oven turvakoodin toteutus

Ensimmäisen päätelaitteen tehtävänä oli huolehtia numeronäppäimistölle syötettävästä nelinumeroisesta tunnusluvusta. Multiprogrammeriin porttiin B yhdistettiin kaksirivinen LCD-näyttö, porttiin C ZigBee päätelaite-kortti ja porttiin D numeronäppäimistö.

Koordinaattori lähetti päätelaitteelle tietoa onko ovi kiinni vai auki, ja tieto tulostettiin LCD-näytölle. Kun käyttäjä painoi ”#”-näppäimen hetkeksi pohjaan, päätelaite kysyi turvakoodia. Mikäli näppäilty nelinumeroinen tunnusluku oli oikein, laite tulosti näytölleen tekstin ”Correct password” ja mikäli tunnus oli väärin, laite tulosti tekstin ”Wrong password”. Päätelaite myös lähetti tunnusluvun tuloksen koordinaattorille, joka puolestaan päätti avataanko ovi vai pidetäänkö ovi kiinni.

Päätelaite ohjelmoitiin käynnistyessään yhdistämään koordinaattoriin, jonka jälkeen ohjelma siirtyi silmukan sisälle varsinaiseen ohjelmaan. Silmukan sisässä ohjelma ensin kuunteli lähettääkö koordinaattori tietoa oven tilasta ja sen jälkeen tarkisti, painetaanko numeronäppäimistöltä painiketta ”#”. Mikäli näppäintä ”#” painettiin, laite pyysi LCD-näytön avulla käyttäjää näppäilemään nelinumeroisen tunnusluvun. Kun tunnusluku oli näppäilty, ohjelma vertasi näppäiltyä tunnuslukua muistiin tallennettuun oikeaan tunnuslukuun. Mikäli näppäilty tunnusluku vastasi oikeaa lukua, lähetettiin koordinaattorille viestinä luku 106 ja mikäli tunnusluku oli väärin, lähetettiin luku 105. Tämän jälkeen ohjelma jäi odottamaan koordinaattorilta kuittauksena lukua 233 merkiksi siitä, että viesti oli vastaanotettu. Kyseiset ohjelmassa käytetyt kokonaisluvut olivat itse keksittyjä lukuja, joiden merkitys vain ohjelmoitiin koordinaattorin ohjelmistoon.

Vaikka päätelaite ei vastaanottanut koordinaattorilta tietoa oven tilasta eikä näppäimistöltä painettu ”#”-näppäintä, lähetettiin koordinaattorille joka tapauksessa noin viiden sekunnin välein numero 137. Tässä tapauksessa numero 137 oli vain merkinä koordinaattorille siitä, että päätelaitteessa on virta päällä ja yhteys on kunnossa.

Päätelaitteen 1 täydellinen vuokaavio on liitteenä 3.

4.4.3 Oven moottoriohjauksen toteutus

Toisen päätelaitteen tehtävänä oli hallita kuvitteellisen oven moottoria. Päätelaitteen multiprogrammerin A-porttiin kytkettiin kaksirivinen LCD-näyttö, B-porttiin neljän releen relekortti, C-porttiin ZigBee päätelaite-kortti ja D-porttiin screw terminal-kortti.

Oven moottoriohjaus toteutettiin liittämällä relekortti ohjaamaan moottorin toimintaa ja suuntaa, sekä liittämällä oven rajakytkimet (RK1 sekä RK2) screw terminal-korttiin ilmoittamaan mikrokontrollerille missä asennossa ovi on. Moottoriin kiinnitettiin moottorin mukana kääntyvä viisari joka osui eri asennoissa rajakytkimiin. Rajakytkimet simuloivat oven auki- ja kiinni-asentoa viisarin osuessa niihin. (kuva 24)



Kuva 24. Oven moottori sekä rajakytkimet

Moottoria ohjattiin relekytkimellä, jonka ensimmäinen rele vastasi moottorin virrasta ja toinen rele moottorin suunnasta. Relekortin jäljelle jääneitä kahta relettä ei käytetty. Releitä hallittiin lähettämällä relekortin porttiin kokonaislukuja, joiden binäärimuotojen ensimmäinen bitti vaikuttaa ensimmäiseen releeseen ja toinen bitti toiseen releeseen. Taulukossa 1 on kuvattu relekortille lähetetyt kokonaislukukomennot ja niitä vastaavat toiminnot.

Taulukko 1. Relekortin moottorin ohjauksen komennot

Tavu binäärimuodossa	Tavu vastaavana kokonaislukuna	Moottorin suunta	Moottorin virta
00000000	0	Kiinni	Off
00000001	1	Kiinni	On
00000010	2	Auki	Off
00000011	3	Auki	On

Relekortti toimii Flowcodessa kuten esimerkkiharjoituksissa käytetty ledi-kortti, joten ohjelmaan lisättiin varsinaisen relekortin sijasta ledi-kortti ja käytettiin ulostulona relekortin porttia.

Oven tilaa tarkkailevat rajakytkimet yhdistettiin multiprogrammeriin screw terminal-kortin avulla. Screw terminal-kortissa on kahdeksan liitännästä sekä maadoitusnastat. Liitännästäisiin syötetyt 5V jännitteet luetaan mikrokontrollerin ymmärtämäksi kahdeksan bitin tavuksi ja kortin toiminta vastasi Flowcodessa aiemmin ohjelmointiesimerkeissä käytettyä painonappien toimintaa ja ohjelmointilogiikkaa. Screw terminal kortilta tulevan tavun ja sitä vastaavan oven tila on kuvattu taulukossa 2.

Taulukko 2. Screw terminal-kortin rajakytkimien ilmoittamat oven tilat

Tavu binäärimuodossa	Tavu vastaavana kokonaislukuna	Oven tila
00000000	0	Puolivälissä
00000001	1	Kiinni
00000010	2	Auki
00000011	3	ERROR

Kun päätelaitteen ohjelma alkoi, ohjelma tarkasti ensimmäisenä oven tilan. Turvallisuussyistä ovi avattiin auki-asentoon mikäli ovi oli kiinni, tai ovi oli vain osittain auki. Ohjelma lähetti relekortin porttiin numeron 3, ja moottori käynnistyi auki-suuntaan. Moottori pysyi käynnissä, kunnes auki-asennon rajakytkin lähetti tiedon että ovi oli avautunut. Tämän jälkeen ohjelma käynnisti ZigBee-verkon, yhdisti koordinaattoriin ja jatkoi silmukan sisälle varsinaiseen ohjelmaan.

Varsinaisessa ohjelmassa päätelaite kuunteli lähettääkö koordinaattori käskyä avata tai sulkea ovi. Mikäli käskyä ei havaittu, lähetti päätelaite koordinaattorille tiedon oven nykyisestä tilasta noin viiden sekunnin välein. Päätelaite lähetti ensin node-numeronsa, sitten oven tilan ja lopuksi näiden yhteenlasketun pariteettitavun.

Mikäli päätelaite vastaanotti lähetyksen, ensimmäisenä ohjelma varmisti että lähettäjänä on koordinaattori. Mikäli lähettäjä oli oikea, päätelaite vastaanotti koordinaattorilta käsky-tavun sekä pariteettitavun ja tarkisti pariteetin avulla vastaanotettiinkö lähetys oikein. Mikäli paketit vastaanotettiin oikein, päätelaite kuittasi koordinaattorille että vastaanotto onnistui, jonka jälkeen ohjelma tarkisti käskikö koordinaattori aukaista vai sulkea oven. Mikäli vastaanotettu käsky oli mahdollista toteuttaa, ohjelma käänsi oven toivottuun asentoon ja lähetti koordinaattorille oven uuden tilatiedon.

Oven moottoriohjauksen vuokaavion toteutus on liitteenä 4.

4.5 Binääriluvun lähetys koordinaattorille

Kolmannen päätelaitteen multiprogrammerin B-porttiin liitettiin ledi-kortti, C-porttiin ZigBee päätelaite-kortti ja D-porttiin painonappikortti. Päätelaitteessa käytettiin samankaltaista toteutusta kuin aiemmin esitellyssä ZigBee-harjoituksessa ”Ledien sytytys painonapeista”. Lähetettävä 8-bitin tavu kerättiin painonapeilta ja tulos lähetettiin sekä päätelaitteen led-kortille sekä koordinaattorille.

Ensimmäiseksi ohjelmassa määriteltiin kaksi uutta byte-tyyppistä muuttujaa. Päätelaitteen Node-muuttujan arvoksi asetettiin 3 ja muuttujan coordinator arvoksi 1.

Seuraavaksi ohjelma käynnisti verkon ja siirtyi silmukan sisälle yhdistämään koordinaattoriin. Kun yhteys oli muodostettu, ohjelma siirtyi varsinaiseen pääohjelmaan silmukan sisälle.

Silmukassa ohjelma luki ensin painonappien tilan ja tallensi arvon muuttujaan ”input”. Tämän jälkeen ohjelma lähetti input-muuttujan arvon ledeille ja laski pariteetin alustuksissa määritetyn node-numeron ja painikkeilta tallennetun input-arvon summasta. Lopuksi päätelaite lähetti koordinaattorille kolme tavua, ensimmäisenä oman node-numeronsa, sitten painonappien input-arvon ja lopuksi lasketun pariteettitavun.

Kerätty binääriluku olisi voitu toteuttaa myös käyttämällä hyväksi screw-terminal -korttia tuoden tietoa esimerkiksi rajakytkimiltä kuten toisen päätelaitteen toteutuksessa tehtiin. Varsinaiseen pääohjelmaan toteutettiin myös ylimääräinen silmukka, jonka sisällä painonappien lukeminen tapahtui. Ylimääräisen silmukan tehtävänä oli pidentää koordinaattorille lähetettävän painonappien tilan ilmoitusväliä, ellei painonapeissa tapahtunut muutoksia. Mikäli painonappeja ei painettu, päätelaite lähetti tiedon tilastaan koordinaattorille noin 5 sekunnin välein ja mikäli painonappien tila muuttui, uusi tilatieto lähetettiin noin 500ms välein.

Binääriluvun lähetyksen täydellinen vuokaavion toteutus on liitteenä 5.

4.6 Lämpötila-anturin toteutus

Lämpötila-anturi toteutettiin liittämällä neljännen päätelaitteen multiprogrammerin A-porttiin sensor board-kortti, B-porttiin LCD-näyttö ja C-porttiin ZigBee päätelaite-kortti.

Neljännen päätelaitteen ohjelmiston toteutus aloitettiin ensin alustamalla tarvittavat muuttujat, kuten laitteen nodenumero. Sen jälkeen ohjelma alusti LCD-näytön, käynnisti ZigBee-verkon ja yhdisti koordinaattoriin. Tämän jälkeen ohjelma siirtyi silmukkaan seuraamaan lämpötilan arvoa.

Lämpötilan lukeminen toteutettiin sensor board-kortin potentiometrillä lisäämällä Flowcoodeen potentiometri ja asettamalla komponentin ”connections”-asetuksista sisääntuloksi ACD: ACD1. Mikäli työssä oltaisiin käytetty potentiometrin sijasta analogista lämpötila-anturia, sisääntuloksi tulisi määrittää ACD: ACD3. Muuten ohjelman toteutus olisi täysin samanlainen kuin potentiometrillä suoritettu toteutus.

Lämpötilatiedon lukeminen tapahtui käyttämällä potentiometrin komponenttimakroa ACD(0): ReadAsByte ja tallentamalla luettu arvo byte-tyyppiseen muuttujaan Temp. Ohjelma vertasi luettua arvoa edelliseen Temp-arvoon, ja mikäli lämpötila-arvo ei ollut muuttunut, keeplalive-muuttujaa kasvatettiin yhdellä ja lämpötila luettiin lyhyen viiveen jälkeen uudelleen. Kun keeplalive-muuttuja saavutti lukeman 10, ohjelma jatkoi eteenpäin ja lähetti koordinaattorille sen hetkisen lämpötilalukeman. Mikäli ohjelma havaitsi lämpötila-arvon muuttuvan, uusi arvo lähetettiin koordinaattorille välittömästi.

Virhearvojen minimoimiseksi potentiometriltä luettu arvo 255 tallennettiin arvoksi 254. Tällöin koordinaattorille lähetetyn lämpötilan arvo saattoi vaihdella 0-254 välillä.

Päätelaite lähetti kerrallaan kolme peräkkäistä tavua koordinaattorille, joista ensimmäinen sisälsi päätelaitteen node-numeron, toinen tallennetun Temp-muuttujan ja viimeinen tavu oli kahdesta edellisestä tavusta yhteenlaskettu pariteettitavu.

Neljännen päätelaitteen vuokaavion täydellinen toteutus on liitteenä 6.

4.7 Verkon kantomatka, luotettavuus sekä analysointi

Työssä käytetyt ZigBee-korttien MaxStreamin valmistamat XBEEV2 ZigBee-piirit ovat speksattu toimimaan sisätiloissa jopa 30 metrin etäisyydellä ja ulkotiloissa jopa 100 metrin etäisyydellä toisistaan. ZigBee-piirin datalehden mukaan niiden lähetysteho on 1mW, ne toimivat 2,4GHz taajuudella ja kykenevät jopa 250kbps tiedonsiirtonopeuteen. Sirujen herkkyys on -92dBm 1% virheellä. /20/

4.7.1 Verkon kantaman mittaukset

Konelaboratorion tiloissa toteutetut verkon kantaman mittaukset antoivat lupaavia tuloksia. Työssä toteutetun verkon laitteet toimivat ongelmitta, vaikka päätelaitteet sijoitettiin laboratorion tiloissa mahdollisimman kauas toisistaan. Konelaboratorion käytävällä suoritettussa kantamatestissä päätelaitteen ja koordinaattorin välinen yhteys säilyi yli 30 metrin käytävän mitalta, mutta mikäli laitteiden välinen etäisyys oli yli 35 metriä ja välistä suljettiin palo-ovi, yhteys menetettiin. Yhteys palo-oven läpi onnistui, mikäli päätelaitteen ja koordinaattorin välinen etäisyys oli noin 10 metriä.

Verkon kantama paranisi todennäköisesti huomattavasti mikäli käytettyyn ZigBee-piiriin liittäisi ulkoisen antennin, sillä työssä käytetyt ZigBee-piirit eivät sisältäneet erillistä antennia. Piirin datalehden perusteella luvattu sisätilan 30m kantama kuitenkin toteutui ainakin esteettömällä käytävällä. Mikäli laitteilla olisi tarkoitus muodostaa sisätiloissa laajempi verkko, reitittimien käyttö olisi välttämätöntä verkon kantaman laajentamiseksi.

4.7.2 Laitteiston luotettavuus ja havaitut ongelmat

Alussa Flowcode -ohjelmaan perehtymisessä ongelmia tuotti ohjelmoinnin logiikan opettelu, mutta yrityksen ja erehdyksen kautta logiikan oppi kuitenkin nopeasti. Usein aikaa ongelmatilanteissa kului liitännöiden tarkistamiseen, korttien vaihtoon sekä ohjelmoidun ohjelman tarkastamiseen. Aika ajoin simulaatiossa hyvin toimiva ohjelma ei toiminut samalla tavalla laitteistossa, ja usein ongelmat johtuivatkin viiveiden puuttumisesta ohjelmarungossa. Mikrokontrolleri kykenee suorittamaan ohjelmarungon eri vaiheet huomattavasti nopeammin kuin tietokone kykenee simuloimaan ohjelmaa.

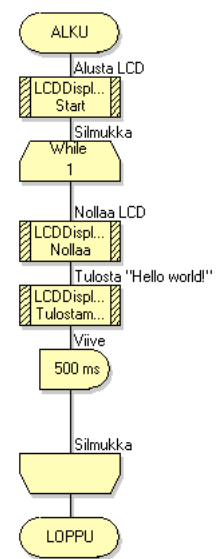
Viiveiden merkityksen ohjelmarungossa voi havainnollistaa esimerkiksi kuvan 25 mukaisessa yksinkertaisessa ohjelmassa. Mikäli viiveen unohtaa lisätä silmukan loppuun, simuloinnissa ohjelma näyttäisi toimivan oikein ja ohjelma tulostaa LCD-näytölle lauseen ”Hello world!”. Varsinaisessa laitteistossa LCD-näyttö kuitenkin näyttää pysyvän pimeänä, sillä näytön nestekiteet eivät ehdi kääntyä tarpeeksi nopeasti kun ohjelma käskyttää kiteitä vuoron perään ensin nollaamaan näytön ja sitten kirjoittamaan lauseen.

Monimutkaisempien ohjelmarunkojen suunnittelu Flowcoden perehtymisen alkuvaiheilla vaativat useita yrityksen ja erehdyksen kaltaisia kokeiluja. Usein ohjelmarungon toteutuksen edetessä ohjelmaa tuli testattua laitteilla aika ajoin, jotta pystyi varmistumaan että ohjelmisto toimi kuten oli tarkoitus. Mitä enemmän Flowcodeen oli perehtynyt, sitä enemmän ohjelmaa uskalsi ohjelmoida kerrallaan ennen toiminnan tarkastusta.

E-Blocks -laitteiston luotettavuuden suurimmat ongelmat ilmentyivät multiprogrammerin ja lisäkorttien välisiin kosketushäiriöihin. Tutkintotyön aikana myös yksi ZigBee päätelaite-kortti vioittui, eikä suostunut toimimaan oikein. Vian paikallistaminen päätelaitekorttiin vei paljon aikaa, mutta kortin vaihtaminen uuteen ratkaisi ongelmat.

Myös opetuslaitteiston multiprogrammerin sekä korttien tarvitseman lisävirran kanssa oli syytä olla tarkkana. Useasti laitteiston toimimattomuus kohdentui käyttäjän omaan huolimattomuuteen. Mikäli esimerkiksi painonappikortin tarvitseman lisävirran unohti kytkeä multiprogrammerilta, aikaa tuhlaantui tarpeettomasti käyttäjän etsiessä vikaa ohjelmarungosta vaikka vika oli kytkennöissä.

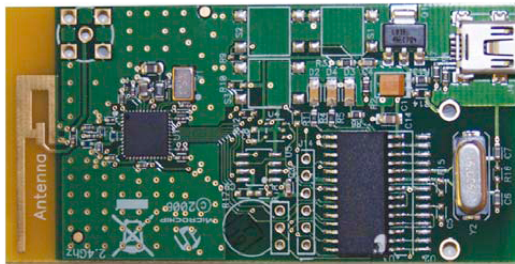
Mikäli opetuslaitteistossa aika ajoin havaittuja kosketushäiriöitä ja muutamia lisäkorttien rikkoutumisia ei oteta huomioon, E-Blocks -sarjan laitteet toimivat hyvin. Laitteiston avulla voi Flowcode-ohjelmaan perehtynyt luoda nopeasti monimutkaisia toimivia laitedemoja. E-Blocks -sarjalla tehdyt demotyöt on kuitenkin syytä valmistaa sulautetuiksi tuotteiksi lopullista käyttöä varten.



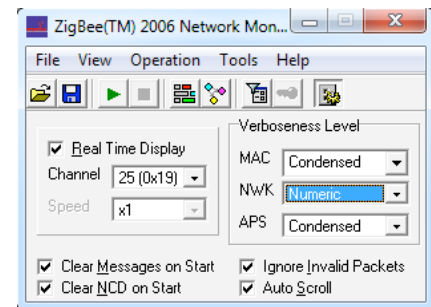
Kuva 25.
Viiveiden merkitys
ohjelmarungossa

4.8 Zigbee-pakettien analysointi ZENA-analysaattorilla

Matrix Multimedian ZigBee -opetussarjaan kuului usb-liitännällä tietokoneeseen liitettävä ZigBee-verkon ZENA-analysaattori, joka on esitelty kuvassa 26. Analysaattorin avulla ZigBee-verkon liikennettä pystyi seuraamaan ja tallentamaan analysaattorin mukana tulleen ohjelman avulla, joka on esitelty kuvassa 27.



Kuva 26. ZENA-analysaattori /21/



Kuva 27. ZENA-analysaattoriohjelma

ZENA-analysaattorin avulla tarkasteltiin esimerkkiharjoituksessa 3.3. muodostetun koordinaattorin sekä päätelaitteen välistä liikennettä. Esimerkissä muodostettiin päätelaitteen ja koordinaattorin välille ZigBee-yhteys, jossa päätelaitteen painonappien tila lähetettiin koordinaattorin led-kortille.

Kuvassa 28 on esimerkkikaappaus tilanteesta kun päätelaite lähettää koordinaattorille tiedon, että painonappien tilan input-muuttujaan tallennettu arvo on 1. Lähetetty arvo näkyy APS Payloadin ikkunassa viimeisenä heksalukuna 0x01. Kun koordinaattori oli saanut paketin vastaan, kuittasi se heti päätelaitteelle että paketti vastaanotettiin.

Kaapatusta paketista voimme todeta esimerkiksi lähettäjän "Source Addr"-ikkunasta, ja vastaanottajan "Dest Addr"-ikkunasta.

Frame	Time(us)	Len	MAC Header	NWK Frame Control	Dest Addr	Source Addr	Radius	Seq Num	APS Header	APS Payload	FCS
00133	+993280 +33664272	45	0x61 0x88 0xFD 0x34 0x02 0x00 0x00 0xE4 0x69	0x00 0x44	0x0000	0x69E4	0x20	0x4A	0x00 0xE8 0x11 0x05 0xE9 0xE1 0xCE	0x00 0xA6 0xB8 0x3A 0x40 0x00 0xA2 0x13 0x00 0x00 0x13 0xA2 0x00 0x40 0x31 0xF7 0x2A 0x01	0xFBEB
Päätelaitteen lähetyk koordinaattorille, datana binääriluku 00000001 eli heksalukuna 0x01											
Frame	Time(us)	Len	MAC Header	NWK Frame Control	Dest Addr	Source Addr	Radius	Seq Num	APS Header	APS Payload	FCS
00134	+2576 +3366848	5	0x02 0x00 0xFD 0xEDEB								
00135	+1824 +33668672	28	0x61 0x88 0x3C 0x34 0x02 0xE4 0x69 0x00 0x00	0x00 0x04	0x69E4	0x0000	0x20	0x34	0x00 0xE8 0x00 0x05 0xE9 0xE1 0xCE	0x04	0xEFEA
Koordinaattori kuittaa päätelaitteelle vastaanottaneensa paketin											
Frame	Time(us)	Len	MAC Header								
00136	+1680 +33670352	5	0x02 0x00 0x3C 0xFBEB								

Kuva 28. Zena-analysaattorilla kaapattu lähetyk päätelaitteelta koordinaattorille.

5 LOPPUSANAT

Työn tavoitteena oli tutustua Matrix Multimedian E-Blocksin ZigBee -opetussarjaan sekä yrityksen tarjoamaan mikrokontrollerien ohjelmointiin tarkoitettuun Flowcode-ohjelmointityökaluun. Työssä tutkittiin kuinka E-Blocks sarjan ZigBee-korttien välille muodostetaan verkko ja kuinka ZigBee-tiedonsiirtoa voitaisiin hyödyntää käytännössä.

Työssä suunniteltiin ja toteutettiin viiden laitteen ZigBee-verkko, jossa neljä päätelaitetta lähettivät tietoa keskuslaitteelle, koordinaattorille. Laitteiston avulla esiteltiin kuinka ZigBeetä voidaan hyödyntää esimerkiksi lämpötila- ja binääritiedon keräämiseen sekä toteutettiin langaton oven avausmoottorin ohjaus kun käyttäjä näppäili oikean tunnusluvun näppäimistölle.

E-Blocks -sarja sekä Flowcode-ohjelmointityökalu osoittautuivat varsin monipuoliseksi kokonaisuudeksi, jonka avulla myös kehittyneeseen ohjelmointiin perehtymätön henkilö pystyy suunnittelemaan ja toteuttamaan monimutkaisiakin laitekokonaisuuksia. E-Blocks -sarjaan on saatavilla monipuolisesti erilaisia yhteensopivia anturikortteja ja sarjan vahvuuksiin kuuluvat nopeasti muodostettavat ja ohjelmoitavat laitteistot. Suurimmat heikoudet liittyivät E-Blocks -sarjan multiprogrammerin ja siihen liitettävien korttien välisiin kosketushäiriöihin, jotka aiheuttivat päänvaivaa suunnittelun ja toteutuksen aikana.

Työn lopputuloksena voidaan todeta, että toteutettu viiden laitteen laitteistokokonaisuus toimi alkuperäisen suunnitelman mukaisesti. Työn toteutuksen edetessä kehittyi jatkuvasti myös uusia ideoita ja mahdollisuuksia mitä laitteistolla olisi voinut toteuttaa, mutta tutkintotyön aiheen rajauksen vuoksi kaikkia ideoita ei ollut mahdollista toteuttaa.

Tutkintotyön tekeminen lisäsi huomattavasti ymmärrystäni ZigBee-verkoista, mikrokontrollerien tarjoamista mahdollisuuksista, E-Blocks -sarjasta sekä Flowcode-ohjelmointityökalusta. Kiinnostukseni E-Blocks -sarjaa ja sen tarjoamia mahdollisuuksia kohtaan jäi myös tutkintotyössä toteutetun laitteiston jälkeen, joten uskon hyödyntäväni tutkintotyössäni oppimiani asioita myös tulevaisuudessa .

LÄHTEET

1. Matrix Multimedia [www-sivu] About [viitattu 7.3.2010] Saatavissa:
<http://www.matrixmultimedia.com/about.php>
2. Matrix Multimedia [www-sivu] E-Blocks hardware [viitattu 7.3.2010] Saatavissa:
<http://www.matrixmultimedia.com/abouteblocks-X.php?C1=Browse%20All%20Products&CAT=E-Blocks%20hardware>
3. Matrix Multimedia [pdf] E-Blocks Multiprogrammer datasheet [viitattu 7.3.2010] Saatavissa: <http://www.matrixmultimedia.com/datasheets/EB006-30-7.pdf>
4. Matrix Multimedia [kuva] E-Blocks [viitattu 7.3.2010] Saatavissa:
<http://www.matrixmultimedia.com/images/system%20shot-7x-1.jpg>
5. Matrix Multimedia [pdf] E-Blocks LCD board datasheet[viitattu 7.3.2010] Saatavissa:
<http://www.matrixmultimedia.com/datasheets/EB005-30-3.pdf>
6. Matrix Multimedia [pdf] E-Blocks graphical LCD display datasheet[viitattu 7.3.2010] Saatavissa: <http://www.matrixmultimedia.com/datasheets/EB043-30-1.pdf>
7. Matrix Multimedia [pdf] E-Blocks relay board [viitattu 7.3.2010] Saatavissa:
<http://www.matrixmultimedia.com/datasheets/EB038-30-1.pdf>
8. Matrix Multimedia [pdf] E-Blocks Zigbee board – Co-ordinator datasheet[viitattu 7.3.2010] Saatavissa: <http://www.matrixmultimedia.com/datasheets/EB051-30-1.pdf>
9. Matrix Multimedia [pdf] E-Blocks screw terminal board datasheet[viitattu 7.3.2010] Saatavissa: <http://www.matrixmultimedia.com/datasheets/EB002-30-1.pdf>
10. Matrix Multimedia [pdf] E-Blocks sensor interface datasheet[viitattu 7.3.2010] Saatavissa: <http://www.matrixmultimedia.com/datasheets/EB003-30-2.pdf>
11. Matrix Multimedia [pdf] E-Blocks keypad datasheet[viitattu 7.3.2010] Saatavissa:
<http://www.matrixmultimedia.com/datasheets/EB014-30-1.pdf>
12. Matrix Multimedia [pdf] E-Blocks switch board datasheet[viitattu 7.3.2010] Saatavissa: <http://www.matrixmultimedia.com/datasheets/EB007-30-1.pdf>

13. Matrix Multimedia [pdf] E-Blocks LED board datasheet[viitattu 7.3.2010] Saatavissa: <http://www.matrixmultimedia.com/datasheets/EB004-30-2.pdf>
14. Matrix Multimedia [pdf] Flowcode ohjelmoinnin perusteet, sivu 33 [viitattu 11.3.2010] Saatavissa: Flowcode_ohjelmoinnin_perusteet_eKirja.pdf, Argon
15. Matrix Multimedia [pdf] Flowcode ohjelmoinnin perusteet, sivu 23-31 [viitattu 11.3.2010] Saatavissa: Flowcode_ohjelmoinnin_perusteet_eKirja.pdf, Argon
16. Matrix Multimedia [pdf] Flowcode ohjelmoinnin perusteet, sivu 32 [viitattu 11.3.2010] Saatavissa: Flowcode_ohjelmoinnin_perusteet_eKirja.pdf, Argon
17. Wikipedia [www] ZigBee [viitattu 11.3.2010] Saatavissa: <http://fi.wikipedia.org/w/index.php?title=ZigBee&oldid=7636923>
18. Fintronic [pdf] Zigbee [viitattu 11.3.2010] Saatavissa: <http://www.fintronic.fi/download/109.pdf>
19. Matrix Multimedia [pdf] EB538 ZigBee Teachers Course Notes [viitattu 7.3.2010] Saatavissa: http://www.matrixmultimedia.com/LearningCentre/EB538-80-1_Zigbee-unlicensed-version.pdf
20. Matrix Multimedia [pdf] XBee/XBee-PRO 802.15.4 OEM RF Modules [viitattu 7.3.2010] Saatavissa: <http://www.matrixmultimedia.com/datasheets/XBEEV2.pdf>
21. Matrix Multimedia [pdf] EB538 ZigBee Teachers Course Notes, sivu 26 [viitattu 7.3.2010] Saatavissa: http://www.matrixmultimedia.com/LearningCentre/EB538-80-1_Zigbee-unlicensed-version.pdf

LIITTEET

Liite 1: ZigBee-verkon muodostus E-Blocks -sarjalla

Tämän laboratorio-ohjeen tarkoituksena on opastaa, kuinka ZigBee-verkon muodostus sekä tiedon lähetys onnistuu ZigBee-verkossa käyttämällä Matrix Multimedian E-Blocks -sarjan ZigBee-kortteja.

Yleistä tietoa

E-Blocks sarjaan kuuluu kahdenlaisia ZigBee-kortteja, kortteja jotka toimivat koordinaattoreina ja kortteja jotka toimivat joko reitittiminä tai päätelaitteina. E-Blocks -sarjan ZigBee-kortin piirilevyyn on merkitty onko toimiiko kortti koordinaattori- vai reititin / päätelaite -moodissa.

Koordinaattorit ovat verkon isäntiä, joiden tehtävänä on käynnistää verkko ja päättää mitkä muut laitteet saavat yhdistää verkkoon. Reitittimien avulla ZigBee-verkon kantamaa voidaan lisätä ja reitittimien tehtävänä on välittää tietoa verkossa koordinaattorin sekä päätelaitteiden välillä. Päätelaitteet toimivat usein esimerkiksi anturitietojen lähettäjinä. Päätelaitteet yhdistävät koordinaattoriin joko suoraan tai reitittimen välityksellä.

ZigBee-kortit tarvitsevat toimiakseen mikrokontrollerin USART-ominaisuutta, ja sellainen löytyy 16F788A mikrokontrolleria käytettäessä multiprogrammerin C-portista. ZigBee-kortit tulee siis liittää multiprogrammerin C-porttiin, jotta Flowcoden ZigBeen komponenttimakrot toimivat oikein. Flowcodessa lisätään wireless-valikosta uusi ZigBee-laite ja ZigBeen asetuksista Flowcodelle kerrotaan kortin portiksi C ja määritetään esimerkiksi kortin nimi, toimiiko kortti koordinaattorina, reitittimenä vai päätelaitteena ja valitaan verkon mahdolliset kanavat. Tarkemmat tiedot ZigBeen asetuksista löydät myöhemmin tästä laboratorio-ohjeesta.

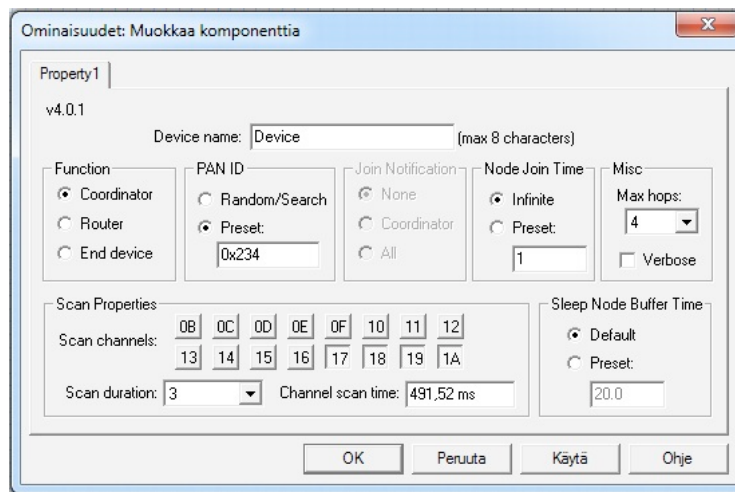
Käytettävät komponenttimakrot Flowcodessa

ZigBee-verkko käynnistetään komponenttimakrolla Init_Network. Tämän jälkeen voidaan määrittää mihin verkon laitteeseen ohjelmoitava laite yhdistää käyttämällä komponenttimakroja Connect_To_Coord, Connect_To_All tai Connect_To_Name –komponenttimakroja. Jälkimmäisen makron parametriksi voidaan asettaa sitaateissa esimerkiksi päätelaitteen nimi, kuten ”End1”. Käytännössä koordinaattorissa voidaan käyttää Connect_To_All-makroa ja päätelaitteissa Connect_To_Coord-makroa.

Tiedonsiirto ZigBee-verkoissa toimii yksinkertaisesti esimerkiksi käyttämällä ZigBeen komponenttimakroa Send_Char ja parametriksi joko yksittäinen numero tai kirjain ’C’, kokonainen sana tai lause sitaateissa ”Hello” tai lähettämällä muuttujan arvo valitsemalla parametriksi aiemmin tallennettu muuttuja ”Muuttujat” valikosta.

Vastaavasti lähetyksen vastaanotto tapahtuu käyttämällä ZigBeen komponenttimakroa ”Reveive_Char”, asettamalla nTimeout parametriksi esimerkiksi arvon 100 ja tallentamalla vastaanotettu tavu käyttäjän valitsemaan muuttujaan.

ZigBee Ext Properties-asetukset Flowcodessa



ZigBee-kortin asetukset

Device name	ZigBee-kortin nimi verkossa
Function	Valitaan toimiiko käytettävä kortti koordinaattorina, reitittimenä vai päätelaitteena.
PAN ID	Yksilöllinen verkon tunniste, jonka avulla verkon laitteet osaavat liittyä samaan ZigBee-verkkoon.
Join notification	Mikäli kortin tehtävänä on toimia reitittimenä tai päätelaitteena, tästä valitaan kenelle kortti ilmoittaa liittyessään verkkoon.
Node Join Time	Valitaan koordinaattorille aika, kauanko verkkoon saa liittyä muita laitteita.
Misc	Valitaan maksimimäärä toistettaville broadcast-lähetyksille ja voidaan kytkeä päälle LCD-näytön tarkan lokin keräys.
Scan Properties	Valitaan kanavat joilta laitteet skannaavat verkkoa sekä aika kuinka kauan kanavia skannataan.
Sleep Node Buffer Time	Valitaan laitteen virransäästöön liittyvät aikamäärytykset.

Zigbee-verkon luominen

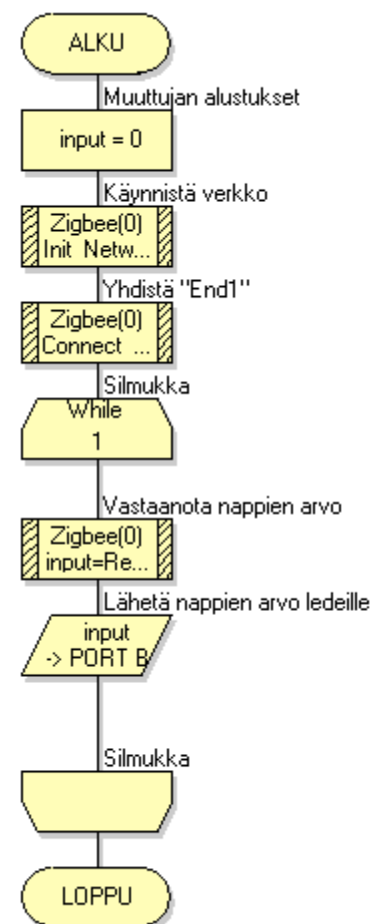
Seuraavassa on esitelty kuinka Flowcodella ohjelmoidaan yksinkertainen verkko kahden laitteen välille ZigBee avulla. Esimerkissä ohjelmoidaan ledien sytytys painonapeista, kun painonapit ja led-kortti sijaitsevat eri multiprogrammereilla ja painonappien arvo lähetetään ledeille ZigBee avulla.

Koordinaattori

Kytke multiprogrammeriin seuraavat kortit

- Porttiin B ledikortti
- Porttiin C ZigBee koordinaattori-kortti

1. Aloita uusi Flowcodetiedosto ja valitse prosessoriksi 16F788A
2. Varmista että ”Prosessori”-valikon kohdasta ”Konfiguroi” oskillaattoriksi on valittu XTAL
3. Lisää outputs-valikosta uusi LEDarray ja määritä sen portiksi B.
4. Lisää Wireless-valikosta uusi Zigbee ja määritä sen portiksi C.
5. Määritä zigbeeen Ext Properties-asetuksiksi
 - Function: Coordinator
 - Device name: Coord
 - PAN ID: preset
 - Join Notification: None
 - Node Join Time: Infinite
 - Misc: Max Hops = 4; Verbose = tyhjä
 - Scan Channels: 17, 18, 19, 1A; Scan Duration = 3; Sleep = Default
6. Lisää ohjelman runkoon uusi laskutoimitus, määrittele uusi byte-tyyppinen muuttuja ja alusta muuttuja nollassa kirjoittamalla laskutoimituksen ikkunaan ”input = 0” ilman sitaatteja.
7. Lisää uusi komponenttimakro ja valitse komponentin Zigbee(0) makroksi Init_Network.
8. Lisää uusi komponenttimakro ja valitse komponentin Zigbee(0) makroksi Connect_To_Name ja parametriksi ”End1” sitaatit mukaan lukien.
9. Lisää ohjelmarunkoon uusi silmukka
10. Lisää silmukan sisälle uusi komponenttimakro ja valitse komponentin Zigbee(0) makroksi Receive_Char, nTimeout parametriksi 100 ja paluuarvoksi muuttuja input.
11. Lisää silmukan sisään ulostulo, jonka asetuksista määritetään muuttujaksi input ja portiksi B.
12. Tallenna harjoitus ja käännä ohjelma mikroprosessorille.



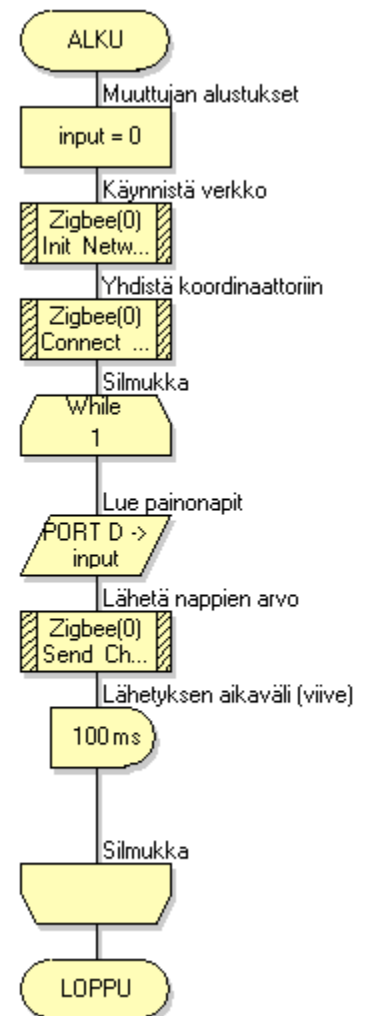
*Koordinaattorin
valmis vuokaavio*

Päätelaite

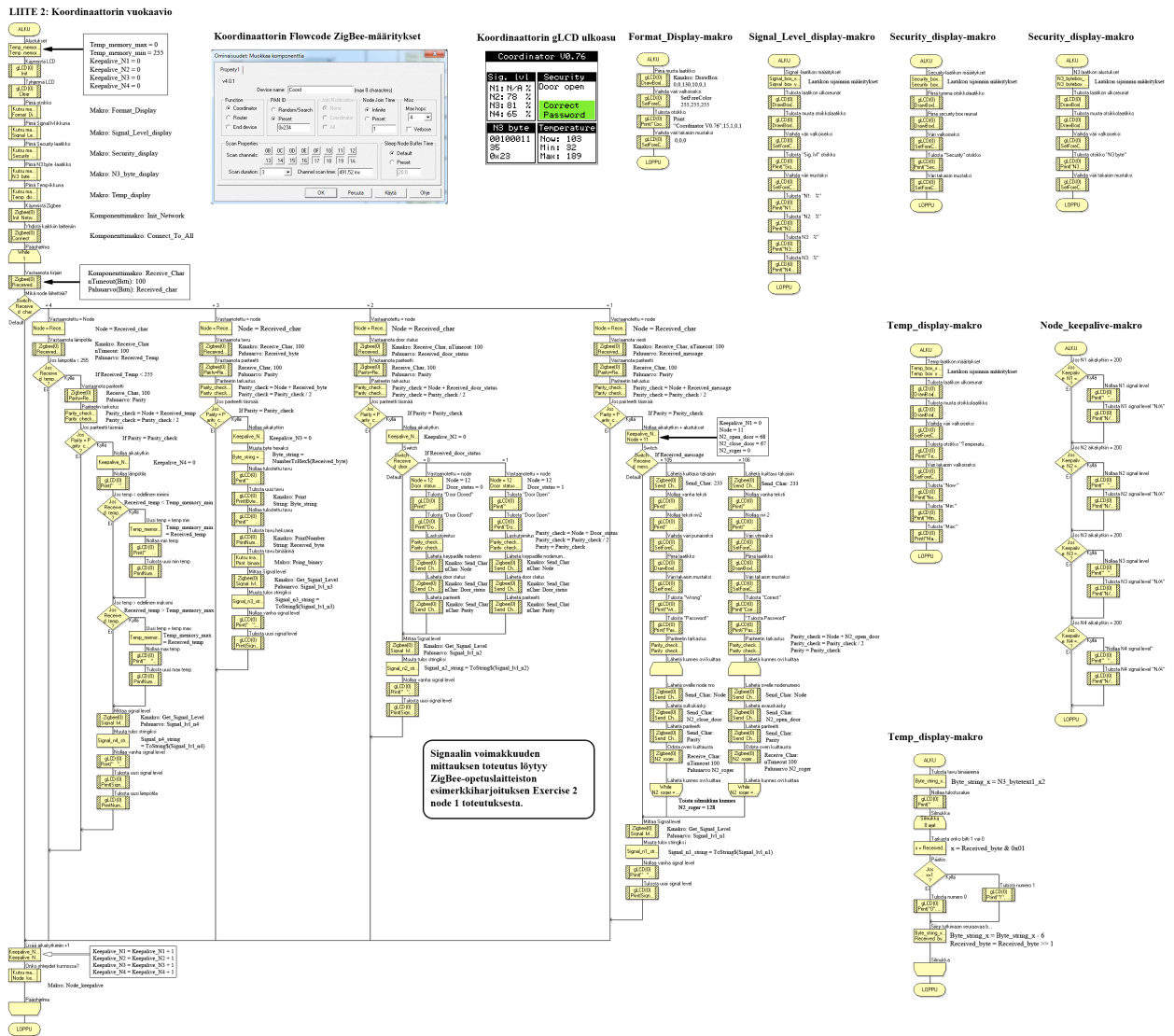
Kytke multiprogrammeriin seuraavat kortit

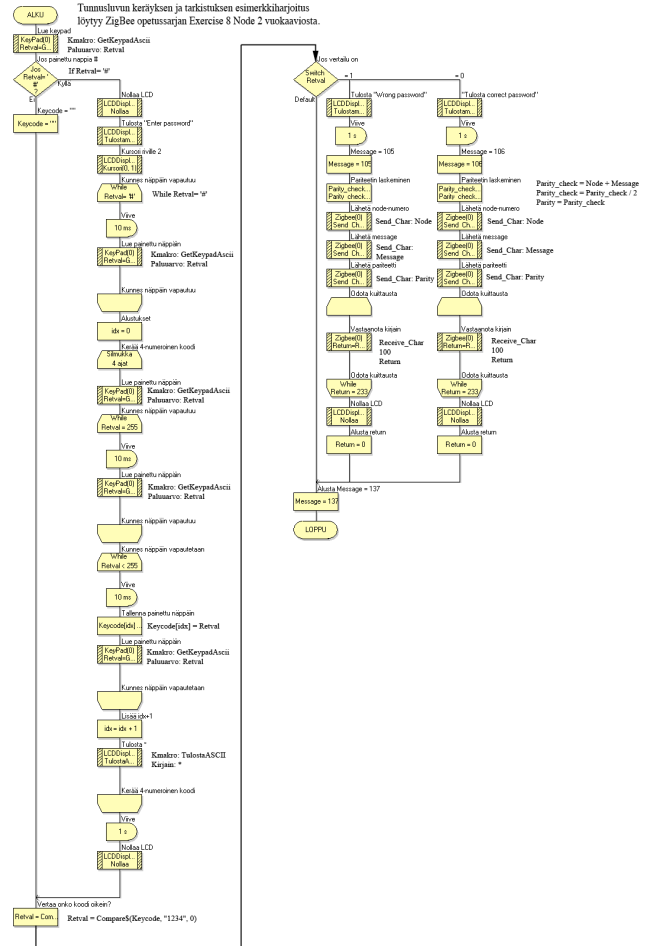
- Porttiin B painonappikortti
- Porttiin C ZigBee päätelaite-kortti

1. Aloita uusi Flowcodetiedosto ja valitse prosessoriksi 16F788A
2. Varmista että ”Prosessori”-valikon kohdasta ”Konfiguroi” oskillaattoriksi on valittu XTAL
3. Lisää inputs-valikosta uusi SWITCHbank ja määritä sen portiksi B.
4. Lisää Wireless-valikosta uusi Zigbee ja määritä sen portiksi C.
5. Määritä zigbeen Ext Properties-asetuksiksi
 - Function: End device
 - Device name: End1
 - PAN ID: preset
 - Join Notification: None
 - Node Join Time: Infinite
 - Misc: Max Hops = 4; Verbose = tyhjä
 - Scan Channels: 17, 18, 19, 1A; Scan Duration = 3; Sleep = Default
6. Lisää ohjelman runkoon uusi laskutoimitus, määrittele uusi byte-tyyppinen muuttuja ja alusta muuttuja nolllaksi kirjoittamalla laskutoimituksen ikkunaan ”input = 0” ilman sitaatteja.
7. Lisää uusi komponenttimakro ja valitse komponentin Zigbee(0) makroksi Init_Network.
8. Lisää uusi komponenttimakro ja valitse komponentin Zigbee(0) makroksi Connect_To_Coord
9. Lisää ohjelmavälikseen uusi silmukka
10. Lisää silmukan sisälle uusi sisääntulo, jonka asetuksista määritetään muuttujaksi input ja portiksi portti D.
11. Lisää silmukan sisälle uusi komponenttimakro ja valitse komponentin Zigbee(0) makroksi Send_Char ja nChar parametriksi muuttuja input.
12. Lisää silmukan sisään 100ms viive.
13. Tallenna harjoitus ja käännä ohjelma mikroprosessorille.



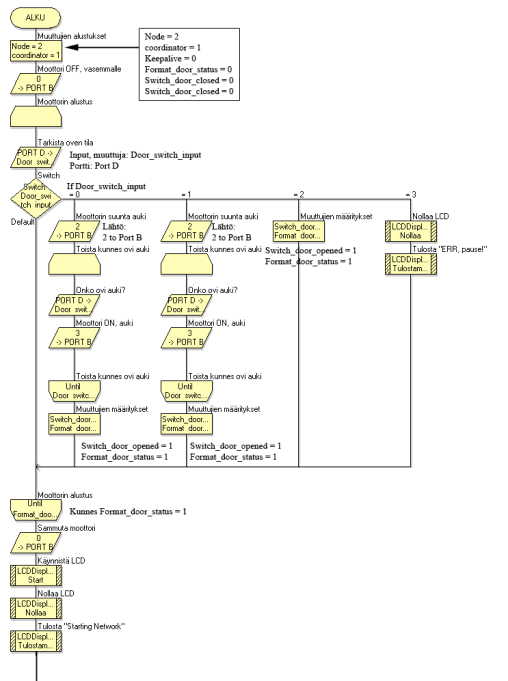
*Päätelaitteen val-
mis vuokaavio*





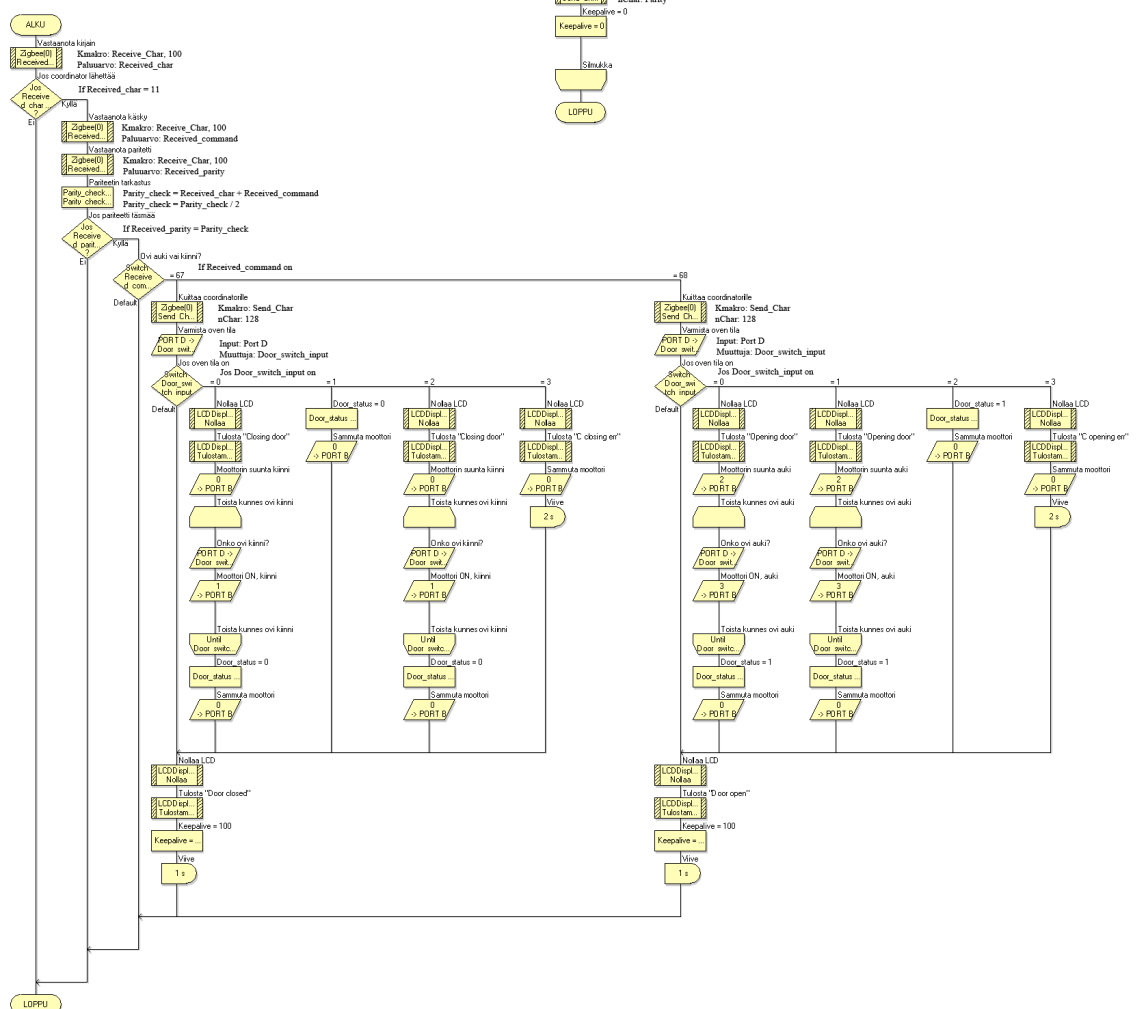
Liite 4: Oven moottorihjauksen vuokaavio

Liite 4: Oven moottorihjauksen vuokaavio



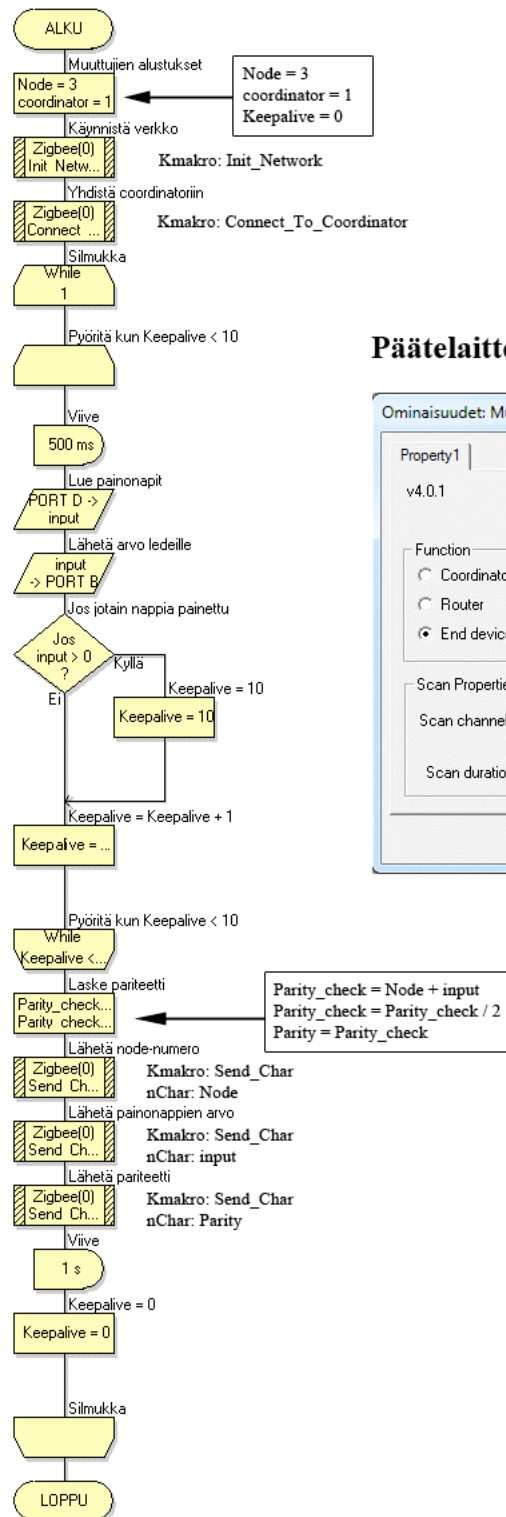
Oven moottorin Flowcoden ZigBee-asetukset

Receive_char-makro



Liite 5: Binääriluvun lähettäjän vuokaavio

Pääohjelma



Päätelaitteen 3 Flowcoden ZigBee-asetukset

Ominaisuudet: Muokkaa komponenttia

Property1

v4.0.1

Device name: Byte (max 8 characters)

Function:

- ☐ Coordinator
- ☐ Router
- ☒ End device

PAN ID:

- ☐ Random/Search
- ☒ Preset: 0x234

Join Notification:

- ☒ None
- ☐ Coordinator
- ☐ All

Node Join Time:

- ☒ Infinite
- ☐ Preset: 1

Misc:

- Max hops: 4
- ☐ Verbose

Scan Properties:

Scan channels: 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A

Scan duration: 3 Channel scan time: 491,52 ms

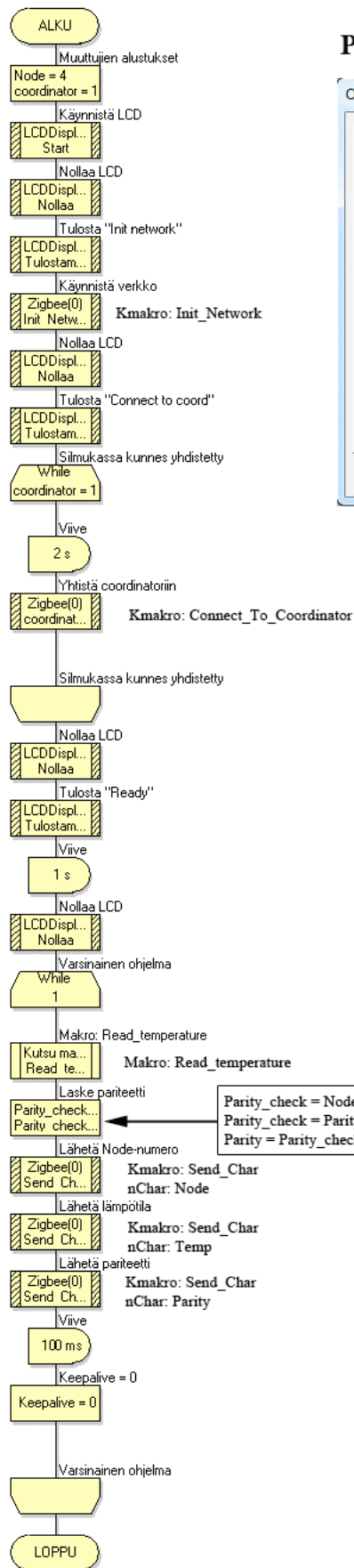
Sleep Node Buffer Time:

- ☒ Default
- ☐ Preset: 20.0

OK Peruuta Käytä Ohje

Liite 6: Lämpötila-anturin vuokaavio

Pääohjelma



Päätelaitteen 4 Flowcoden ZigBee-asetukset

Ominaisuudet: Muokkaa komponenttia

Property1

v4.0.1

Device name: Temp (max 8 characters)

Function:

- ☐ Coordinator
- ☐ Router
- ☒ End device

PAN ID:

- ☐ Random/Search
- ☒ Preset: 0x234

Join Notification:

- ☒ None
- ☐ Coordinator
- ☐ All

Node Join Time:

- ☒ Infinite
- ☐ Preset: 1

Misc:

- Max hops: 4
- ☐ Verbose

Scan Properties:

Scan channels: 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A

Scan duration: 3 Channel scan time: 491,52 ms

Sleep Node Buffer Time:

- ☒ Default
- ☐ Preset: 20.0

OK Peruta Käytä Ohje

Read_temperature-makro

